8-8-2017

FULL-VIEW COVERAGE PROBLEMS IN CAMERA SENSOR NETWORKS

Chaoyang Li
Georgia State University

FULL-VIEW COVERAGE PROBLEMS IN CAMERA SENSOR NETWORKS

by

Chaoyang Li

Under the Direction of Anu G. Bourgeois, PhD

ABSTRACT

Camera Sensor Networks (CSNs) have emerged as an information-rich sensing modality with many potential applications and have received much research attention over the past few years. One of the major challenges in research for CSNs is that camera sensors are different from traditional scalar sensors, as different cameras from different positions can form distinct views of the object in question. As a result, simply combining the sensing range of the cameras across the field does not necessarily form an effective camera coverage, since the face image (or the targeted aspect) of the object may be missed. The angle between the object's facing direction and the camera's viewing direction is used to measure the quality of sensing in CSNs instead. This distinction makes the coverage verification and deployment methodology dedicated to conventional sensor networks unsuitable.

A new coverage model called full-view coverage can precisely characterize the features of coverage in CSNs. An object is full-view covered if there is always a camera to cover it no matter which direction it faces and the camera's viewing direction is sufficiently close to the object's facing direction. In this dissertation, we consider three areas of research for CSNS: 1. an analytical theory for full-view coverage; 2. energy efficiency issues in full-view coverage CSNs; 3. Multi-dimension full-view coverage theory. For the first topic, we propose a novel analytical full-view coverage theory, where the set of full-view covered points is produced by numerical methodology. Based on this theory, we solve the following problems. First, we address the full-view coverage holes detection problem and provide the healing solutions. Second, we propose k-Full-View-Coverage algorithms in camera sensor networks. Finally, we address the camera sensor density minimization problem for triangular lattice based deployment in full-view covered camera sensor networks, where we argue that there is a flaw in the previous literature, and present our corresponding solution. For the second topic, we discuss lifetime and full-view coverage guarantees through distributed algorithms in camera sensor networks. Another energy issue we discuss is about object tracking problems in full-view coverage camera sensor networks. Next, the third topic addresses multi-dimension full-view coverage problem where we propose a novel 3D full-view coverage model, and we tackle the full-view coverage optimization problem in order to minimize the number of camera sensors and demonstrate a valid solution.

This research is important due to the numerous applications for CSNs. Especially some deployment can be in remote locations, it is critical to efficiently obtain accurate meaningful data.

INDEX WORDS: Full-view coverage, Algorithm, Networking, Energy efficiency

FULL-VIEW COVERAGE PROBLEMS IN CAMERA SENSOR NETWORKS

by

Chaoyang Li

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2017

FULL-VIEW COVERAGE PROBLEMS IN CAMERA SENSOR NETWORKS

by

Chaoyang Li

Committee Chair: Anu G. Bourgeois

Committee: Guantao Chen

Yingshu Li

Wei Li

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

August 2017

DEDICATION

To Ouyang Shihong, my wife, without whom, I would not be who I am today.

To Xiaohan Li, my son, who always supports me.

Acknowledgements

I would like to take the chance to thank Dr. Anu G. Bourgeois, my advisor, for guiding me through the PhD study. Thank you for all of the meetings and chats over the years. You recognized that I at times needed to work alone but also made sure to check in on me so that I stayed on the right path. I hope that I could be as lively, enthusiastic, and energetic as you. I always feel amazing about and admiring for your super-high efficiency for handling an enormous number of issues each day, and hope your good habit and methodologies will make me as productive as possible in my future academic life. I also admire and respect your perfect pronunciation and excellent communication skills.

To Tammie Dudley. Thanks for your ever-present sunshine-like smile and quick response when I need your help.

To Dr. Yingshu Li, Dr. Wei Li, Dr.Guantao Chen. I wish to offer my most heartfelt thanks to you. This dissertation could not have been completed without the great support from you.

To Dr. Rajshekhar Sunderraman. As the department's graduate Chair, you have been an ever present beacon of support, encouragement, and advice. Thank you for keeping your office door open and available for all the times when I needed it.

To Yang Wang. I have always been able to turn to you when I have needed a listening ear, a dissenting opinion, or someone to gauge the spiciness of some food. You are great friends and a rock of stability in my life. Our friendship was strong before and I hope it will continue to be as strong in the future.

To my Mom and Dad. You have encouraged my academic interests from day one, even when my curiosity led to incidents that were kind of hard to explain. Thank you.

To anyone that may I have forgotten. I apologize. Thank you as well.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

With the advances in image or video sensor technology, camera sensors have appeared in a number of products such as cell phones, toys, computers and robots and are increasingly employed in many applications such as surveillance, traffic monitoring, habitat monitoring, healthcare and even online gaming [2, 3, 4].

Unlike traditional wireless sensors, which measure scalar phenomena such as acoustic or thermal parameters in the physical world, camera sensors are capable of adaptively detecting, recognizing and tracking intruders (e.g. vehicles or people) within their sensing ranges, recording vivid image shots of a surveillance region, hence retrieving much richer information in the form of images or videos and providing more detailed and interesting data of the environment. Due to the numerous applications, huge research efforts have been focused on the camera sensor networks in the past few years[5, 6, 7, 8, 9].

Wireless sensor networks consist of sensor nodes each capable of sensing environment and transmission data. Broadly speaking, a sensor responds to a physical stimulus such as heat, light, sound, pressure, magnetism, etc. and converts the quality or parameter into electrical signals, which then are digitalized into sensing data. A sensor node usually constitutes one or more sensor units, a power supply unit, a data processing unit, a data storage, and a data transmitting unit. In some cases, one or more sinks are included in sensor networks to collect data from sensor nodes. A large number of sensor nodes then are deployed in different geographical locations within a sensing field to collectively and coordinately monitor physical phenomena.

One fundamental problem is determining how well the target field is monitored, which is referred to as the coverage problem in wireless sensor networks. Most existing works in this field focused on traditional sensor networks and they suggested a very simple model for characterizing the coverage: an object is considered to be covered if it is within the sensor's sensing range, which can be a disk [10] or sector [3]. With this generic model, extensive studies have been devoted to the problem of how to achieve k-coverage over a given target area [11, 12, 2], where k is a predefined parameter indicating the desired

number of sensors (coverage degree) covering each object.

However, camera sensors are different from traditional scalar sensors. Camera sensors may generate very different views of the same object if they are from different viewpoints. For example, a camera sensor placed in front of a person can obtain the face image, but it can only view his back if it is behind him. In fact, studies in computer vision show that the object is more likely to be recognized by the recognition system if the image is captured at or near the frontal viewpoint [13], i.e., if the object is facing straight to the camera. As the angle between the object's facing direction and the camera's viewing direction (denoted by the vector from the object to the camera) increases, the detection rate drops dramatically [4, 14]. As a result, the quality of coverage is determined not only by the distance between the sensor and the object but also by the relationship between the sensor's viewing direction and the object's facing direction. Given the new characteristics of camera sensor networks, how do we define coverage, i.e., what should a "good" camera sensor network be? Is there any efficient method to evaluate the coverage quality of a given camera sensor network? Furthermore, how many camera sensors are sufficient to achieve a "good" coverage in practice, either in a random or a deterministic deployment? All these questions need to be answered before the camera sensor networks can be widely deployed in practice.

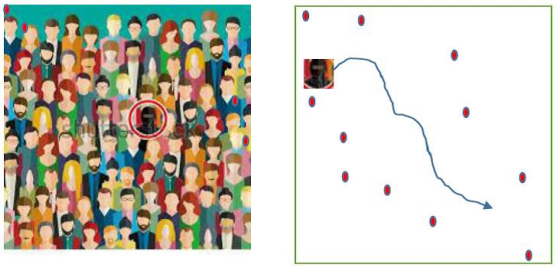

Figure 1.1: Left: a terrorist suspect among people. Right: Terrorist suspect is full-view covered.

In particular, we consider the full-view coverage problem, which attempts to capture full-view coverage images. This is very significant in many applications. For example, it is highly demanded to provide k-full-view coverage for a whole region where a public event occurs. In such situations, a large number of people gather in a space-confined region. If there was a terrorist attack, many people's lives would be endangered and resultant damages would be immeasurable. Further, it may take longer time for the police, even with the help from the present audiences, to find the suspects. Let alone prevent the attack. If we can set up camera sensor networks to provide full-view coverage in such scenarios, we would efficiently identify the suspects and even prevent the attack, as shown in Figure 1.1. In this scenario, full-view coverage can achieve better coverage quality than traditional coverage concept, since for a suspect, we can monitor the potential suspect in all directions by utilizing the full-view coverage technique.

The second problem is about full-view coverage hole detection and healing solution in camera sensor

networks. There are literatures involved with coverage holes detection and healing solution[15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28] in wireless sensor networks. Among many problems confronted in designing camera sensor networks (CSNs), the coverage challenge stands out as one of most fundamental problems. Some CSNs applications, such as environment monitoring and animal tracking, require to cover each point of the target area. However, even if we can deploy camera sensor nodes to make the entire target region fully full-view covered at the very beginning, the nodes can die due to battery drain or environmental causes, which may generate full-view coverage holes in the region. Also, nodes may deviate from their initially assigned locations due to uncontrollable elements such as strong wind, leaving some areas not full-view covered. Full-view coverage holes reduce the ability of CSNs to detect events and network reliability. Consequently, it is critical to equip camera nodes with efficient hole detection and recovery capabilities to ensure the full-view coverage of the whole target field.

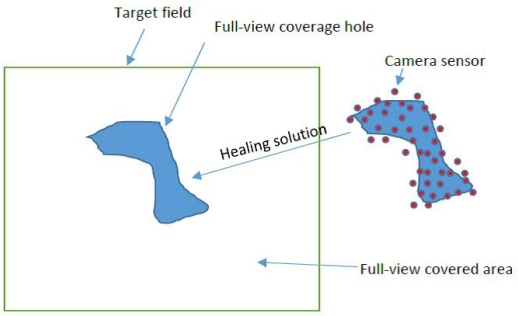

Figure 1.2: The colored region is a full-view coverage hole, and the healing solution provides a patching capability.

As shown in Figure 1.2, the shadowed subregion in the rectangle target area is detected as a full-view coverage hole, where is not full-view covered. In order to patch this subregion, a healing technique is explored, where extra camera sensors are deployed to make it re-full-view covered.

The third problem we focus on is the minimization formula for equilateral triangle lattice density It has been proven that triangular deployment is the best choice to cover a given area with a minimum number of sensors. In this dissertation, we investigate in detail the camera sensor density minimization problem for triangular lattice based deployment in full-view covered camera sensor networks. We prove the formula in the literature [1] which aims to minimize triangular lattice density is not correct. We then demonstrate a minimization formula for equilateral triangle lattice-based deployment and prove its correctness. Finally, we perform simulations and the results show our conclusions are correct. When a vast number of camera sensors are randomly placed in a target field, how can we produce a sleep/activate schedule for camera sensors to maximize the lifetime of target coverage in the target field? This well-known problem is called Maximum Lifetime Coverage Problem (MLCP)[29], which has been explored extensively in the literature. However, MLCP in full-view coverage camera sensor networks is far more complicated since each target should be full-view covered by a group of camera sensors instead of a single

sensor as in conventional scenarios. we address the full-view covered target problem, with the objective of maximizing the lifetime of a power constrained camera sensor network deployed for full-view coverage of targets with a known location.

Object tracking in camera sensor network is a quite representative application which has widespread use in applications such as security surveillance and wildlife habitat monitoring [30]. During tracking period, we expect that camera sensors remain largely inactive for long periods of time but then becoming suddenly active when the moving object which we need to monitor is detected. Tracking involves a significant amount of collaboration between individual camera sensors to perform full-view coverage. This collaborative nature of camera sensor networks offers significant opportunities for energy management. For example, just the decision of whether to do the collaborative full-view coverage at the user end-point or somewhere inside the network has significant implication on energy and lifetime. We focus on the issue of minimization of the power consumed by camera sensor nodes, which involves the object tracking. During the object tracking, some camera sensors located alongside the object moving path will be turned on to achieve full-view coverage goal at the beginning of tracking object and turned off to save energy after the object moves away, the timing of turning on/off a particular camera sensor and the switch of adjacent camera sensors depend on the object's movement and location information. So, in this case, the camera sensor network is not a continuous-connected one. It is designed for intermittent network connectivity. We will develop a brute-force algorithm and a heuristic algorithm to achieve energy optimization goal during the object tracking process in the full-view coverage camera sensor networks.

In some applications, the camera network are distributed over a 3D space, e.g., camera sensors are required to be placed on the trees of different heights in a forest. So it is important to study 3-D full-view coverage.

The remainder of the dissertation is organized as follows. In the fourth chapter, we address the lifetime and full-view coverage guarantees through distributed algorithms in camera sensor networks, we also address object tracking problems to minimization the energy consumed. In chapter 5 we discuss full-view coverage in 3D space. Chapter 6 concludes the dissertation and shows the impact of our work on future research.

Chapter 2

Background

In this chapter, we review the related work and briefly compare our work with those most related.

2.1 k-Coverage Problem

As mentioned before, coverage problem under disk sensing model has been studied extensively in the past few years. More comprehensive surveys on coverage detection (verification) methods can be found in [14, 31]. Here we only review the most relevant work.

K-coverage detection techniques can be classified as follows.

In [32], it is shown that an area is k-covered if and only if the perimeter of all sensor's sensing range (disk) is k-covered. A polynomial-time detection algorithm has been proposed based on this perimeter coverage property.

In [33], the idea of perimeter coverage has been developed into a distributed protocol in which no location but only distance information is assumed to be known by the sensors.

Based on the same assumption, Kasbekar et al. [34] show that the target field is k-covered if the intersection points of the perimeter of any two sensors' sensing disks are k-covered. They also present a distributed protocol which schedules the sensors to prolong the lifetime of the network with coverage guarantee.

Another direction on coverage detection is to utilize the property of the Voronoi Diagram to eliminating redundancy and improve energy efficiency, while preserving the network's coverage. Some interesting works such as [35, 21] etc., verify this statement. Sensor density estimation for k-coverage has been studied in [31, 36]. In [31], three kinds of deployments, square grid, uniform distribution and Poisson distribution, are considered. Each sensor is assumed to work with probability p. Critical condition on the relationship among the number of sensors, sensing radius and the working probability (p) for the target field to be almost surely k-covered has been established.

As people realized that requiring the whole target field to be k-covered (full coverage) is not always necessary, various models on partial coverage are proposed. Barrier coverage [31] and trap coverage [37] are two such variants. In barrier coverage, the sensing disks of the active sensors form a strip zone which serves as a barrier, and any object crossing the target field is supposed to be detected by the sensors on the barrier. In trap coverage, coverage holes are allowed to exist as long as the diameters of the holes are bounded. Sensor density estimations for these coverage requirements are derived [31, 37, 38, 39]. The optimal deterministic deployment pattern for 1-coverage is based on triangle lattices, which has been proved in [40]. One of the latest results on achieving both coverage and connectivity in deterministic deployment under disk model can be found in [41]. With the development of new visual sensor technology, camera sensor networks have received much attention recently in research [10, 11, 12, 4, 42, 43]. One basic problem is how to characterize the usefulness of the image data and how to optimize the network to achieve a better quality of information. However, very little effort has been devoted to this field. One problem studied is called pan and scan [44], which is proposed to maximize the total coverage in a camera sensor networks. For camera sensor placement, various optimization models and heuristics are studied in [3]. However, the coverage model is relatively simple, depending only on the distance between the target and the object, which does not consider the uniqueness of photo coverage. The above studies of disk coverage model and camera sensor networks inspire our work in this proposal. The major difference between theirs and ours is that full-view coverage requires consideration of three factors: the distance between the point and the sensor, the viewing direction of the sensor, and the orientation of the sensor, while in disk model, only the distance needs to be considered. All these issues make the full-view coverage problem far more complicated and challenging, where new strategies, novel theories and algorithms are needed to handle them.

2.2 Object Tracking Problem in Camera Sensor Networks

The demand for surveillance systems has increased rapidly over recent times. In [45], a system consisting of a distributed network of cameras that allows for tracking and handover of multiple persons in real time is presented. An energy-efficient smart camera mote architecture is introduced in [46], where a low-resolution stereo vision system continuously determines position, range, and size of moving objects entering its field of view. The problem tracking many objects with many sensors is investigated in [47]. In [48] an approach for dynamic sensor selection in large video-based sensor networks for the purpose of multi-camera object tracking is presented. The paper [49] describes the theory and implementation of a system of distributed sensors which work together to identify and track moving people using different sensing modalities in real time, where algorithms for detecting people using cameras and laser scanners are presented. Barrier Coverage in Camera Sensor Networks is addressed in [50, 51, 52, 53, 54], where a cost

efficient "camera barrier" is built up such that every intruder's image can be detected effectively. To our best knowledge, no literature involving object tracking in full-view covered camera sensor networks exist. It is important to address object tracking problem where full-view coverage can provide better security guarantee, especially under certain circumstances such as terrorists attack in public area. This problem is more complicated than the conventional object tracking counterparts due to camera's direction-sensitive features.

Chapter 3

Analytical Theory for Full-View Coverage

In this chapter, we introduce the k-full-view coverage model and algorithms. Then we demonstrate full-view coverage analytical theory, based on which we illustrate the full-view coverage hole detection and healing technique. Finally, we show the minimization formula for equilateral triangle lattice-based deployment.

3.1 Full-View Coverage Holes Detection and Healing Solutions

In this section, we address the full-view coverage holes detection problem and healing solutions. Among many problems confronted in designing camera sensor networks (CSNs), the coverage challenge stands out as one of most fundamental problems. Some CSNs applications such as environment monitoring and animal tracking require covering each point of the target area. However, even if we can deploy camera sensor nodes to make the entire target region full-view covered at the very beginning, the nodes can die due to battery drain or environmental causes, which may generate full-view coverage holes in the region. Also, nodes may deviate from their initially assigned locations due to uncontrollable elements, leaving some areas not full-view covered. Full-view coverage holes reduce the ability of CSNs to detect events and network reliability. Consequently, it is critical to equip sensor nodes with efficient hole detection and recovery capabilities to ensure whole full-view coverage of the target field.

We develop a novel full-view coverage analytical theory to address the full-view coverage holes detections problem, where coordinate system is applied to manipulate equations for full-view covered points, curves, and area. On the other hand, full-view coverage holes can be detected by this theory. The hole's boundary can be represented precisely by coordinates and the area of holes can be calculated quantitatively. This theory can also be utilized to study the relationship between the sensing range or the

effective angle of cameras and full-view coverage area, therefore, it can serve as a guideline for designing full-view coverage camera sensor networks.

Besides full-view coverage hole detection, full-view coverage hole recovery is a key issue in the full-view covered CSNs. We develop traditional methods such as using sensor movement to improve network full-view coverage, where sensor nodes are equipped with mobile platforms moving around after the initial deployment. Since the centralized schemes are not feasible in large-scale CSNs because they burden the central node while sensor nodes have limited energy and computation capacity, as a consequence, bottles will be generated. Hence, we explore the possible distributed scheme to solve full-view coverage hole problem.

3.1.1 Analytic Theory of Full-View Coverage

In this section, we present the novel analytic theory of full-view coverage. In our analytical model for full-view coverage, We deduce the equiangular loci corresponding to the arbitrary camera pair based on numerical analysis methodology. Based on the concept of equiangular loci, we present the concept of the full-view covered single connected domain where each point in the domain is full-view covered, and where a full-view covered single point is a special case of the full-view covered single connected domain, where the single connected domain degenerates into a point.

3.1.2 Adjacent Camera Sensor Pair and It's Equiangular Loci

Given an arbitrary point P (target), there exists a camera set $C = (C_1, C_2, ..., C_n)$, where any element $C_i \in C$ is within the sensing range R_s from P. The relative position of C_i to P can be represented by the polar coordinate of C_i, i.e., $C_i = c_i e^{i\theta_i}$, where c_i is the Cartesian distance from camera C_i to target P, and θ_i is the polar angle from a reference direction. Here the target is the pole. If we sort the cameras in C by their angular coordinates, then we can get a list $C_l = (C_1, C_2, ..., C_n)$.

Definition 1 *An adjacent camera pair C_p is a pair (C_k, C_{k+1}) in C_l, where $1 \leq k \leq n$, and $C_{n+1} = C_1$.*

In order to achieve full-view coverage for the target P, for any adjacent camera pair $C_p = (C_k, C_{k+1})$, the following inequality must be satisfied because of the second necessary condition for full-view coverage:

$$\angle C_k P C_{k+1} \leq 2\theta_0, k \in \{1, ..., n\} \tag{3.1}$$

The above inequality is equivalent to:

$$\overrightarrow{PC_k} \cdot \overrightarrow{PC_{k+1}} \geq |PC_{k+1}|\,|PC_k|\,cos(2\theta_0) \tag{3.2}$$

Consequently, the boundary condition is defined as:

$$\overrightarrow{PC_k} \cdot \overrightarrow{PC_{k+1}} = |PC_{k+1}|\,|PC_k|\,cos(2\theta_0) \tag{3.3}$$

Using polar coordinates, the kth cameras' position can be expressed as follows:

$$C_k = c_k e^{i\theta_k} = (c_{kx}, c_{ky}) \tag{3.4}$$

$$C_{k+1} = c_{k+1} e^{i\theta_{k+1}} = (c_{(k+1)x}, c_{(k+1)y}) \tag{3.5}$$

Similarly, any point P on the boundary can be represented as:

$$P = r\theta^{i\theta} = (p_x, p_y) = (rcos\theta, rsin\theta) \tag{3.6}$$

Substituting equations 3.4, 3.5 and 3.6 into equation 3.3, we get:

$$\begin{aligned} &(c_{kx} - rcos\theta)(c_{(k+1)x} - rcos\theta) + \\ &(c_{ky} - rsin\theta)(c_{(k+1)y} - rsin\theta) \\ &= \sqrt{(c_{kx} - rcos\theta)^2 + (c_{ky} - rsin\theta)^2} \\ &\times \sqrt{(c_{(k+1)x} - rcos\theta)^2 + (c_{(k+1)y} - rsin\theta)^2} cos(2\theta_0) \end{aligned} \tag{3.7}$$

Equation 3.7 can be solved numerically, resulting in a solution as shown in Figure 3.1. Without the loss of generality, we indicate the camera pair as C_3 and C_4. Note that this curve is not closed; C_3 and C_4 are two breakpoints. The reason is that the limits on the left and right side of C_3 and C_4 do not exist; the point P on the curve can infinitely approach C_3 and C_4, but can never reach them. For this reason, we exclude these two points during the following discussion.

If point P is on the boundary, then $\angle C_3 P C_4 = 2\theta_0$. If point P_{out} is outside the boundary, then $\angle C_3 P_{out} C_4 < 2\theta_0$. If P_{in} is inside the boundary, then $\angle C_3 P_{in} C_4 > 2\theta_0$.

Proof: For any point P_{in} that is located inside the boundary curve, connect P_{in} and C_3, then extend the resulting segment until it intersects the boundary curve at point P_0. After connecting C_4 and P_0, and C_4 and P_{in}, it is obvious that $\angle C_3 P_{in} C_4 > \angle C_3 P_0 C_4 = 2\theta_0$. For any point P_{out} located outside the boundary curve, we can prove that $\angle C_3 P_{out} C_4 < \angle C_3 P_0 C_4 = 2\theta_0$ by using a similar strategy.

Definition 2 *An adjacent camera pair's exterior region is the area outside of their equiangular loci, including the equiangular loci. An adjacent camera pair's interior region is the area inside their equiangular loci.*

Obviously, given the adjacent camera pair C_k, C_{k+1} and point P that is located within the exterior

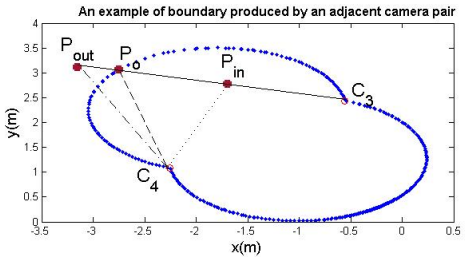

Figure 3.1: The solution for Equation 3.7.

region, we have $\angle C_k P C_{k+1} \leq 2\theta_0$.

On the other hand, if point P is located within the interior region, we have $\angle C_k P C_{k+1} > 2\theta_0$.

3.1.3 Necessary and Sufficient Conditions for Full-View Coverage

Lemma 1 *For an arbitrary point P, and a camera set C, if:*

- *P is within the sensing range of every individual camera sensor in the camera set.*
- *P is within the interior area of any adjacent camera pair C_k and C_{k+1}.*

then P is full-view covered by C.

We call the above two conditions the first and the second necessary conditions for full-view coverage, respectively. These two individually necessary conditions are jointly sufficient for full-view coverage for the point P [1].

Theorem 1 *Suppose the camera set $C = (C_1, C_2, ..., C_n)$ constructs a polygon and the polygon's maximum diameter d is not greater than the sensing range R_s. If the equiangular loci of all adjacent camera pairs constitute a simply connected domain, and this simply connected domain is located within every adjacent camera pair's exterior region, then this area is full-view covered by C.*

Proof: If the maximum diameter $d \leq R_s$, then the first necessary condition for full-view coverage is satisfied, since any point in the simply connected domain is within the sensing range for all cameras in C. If the simply connected domain is located within every adjacent camera pair's exterior region, then the second necessary condition for full-view coverage is also satisfied. As a result, any point in the simply connected domain will be full-view covered by the camera set C.

For example, as illustrated in Figure 3.2, the camera set $C = (C_1, C_2, ..., C_6)$ constitutes a polygon connected by the dashed lines, and the maximum diameter d of the polygon is not greater than the sensing range R_s. The simply connected domain constituted by arcs $C_1C_2, C_2C_3, C_3C_4, C_4C_5, C_5C_6, C_6C_1$ is located within the exterior regions of all adjacent camera pairs. According to Theorem 1, any point in the simply connected domain is full-view covered by C.

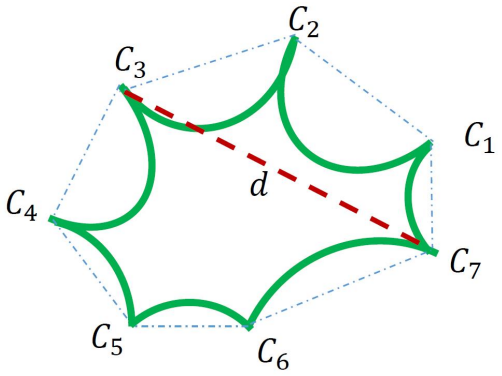

Figure 3.2: The simply connected domain is full-view covered if the sensing range $R_s \geq d$.

Corollary 1 *If the area of the simply connected domain is equal to zero, then the intersection point is full-view covered.*

The corollary 1 is a special case of Theorem 1, in which case the simply connected domain degenerates into a point. It can be used as a criterion to judge whether or not an arbitrary point is full-view covered.

Definition 3 *Minimal full-view coverage camera set C is a camera sensor set that can full-view cover an area, where no subset of C can full-view cover any area.*

It is apparent that the minimum number of camera sensors in the minimal full-view coverage camera set C is $\left\lceil \frac{\pi}{\theta_0} \right\rceil$. This statement can be proven as follows: in order to minimize the number of camera sensors needed, the angle between two adjacent cameras should be maximized, where the maximum value equals to $2\theta_0$. According to the pigeonhole principle, the minimum number of camera sensors in the minimal full-view coverage camera set C is $\left\lceil \frac{\pi}{\theta_0} \right\rceil$.

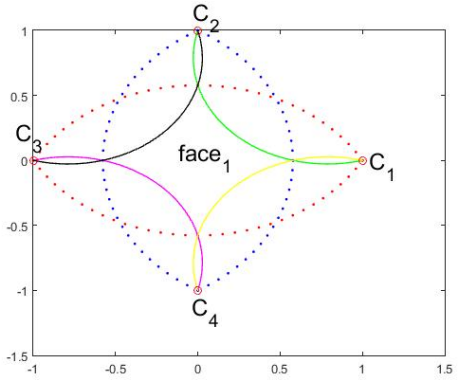

Figure 3.3: An example of minimal full-view coverage camera set.

In Figure 3.3, the sensing range $R_s \geq |C_1C_3|, |C_2C_4|$ and the effective angle $\theta_0 = \pi/3$. By applying Theorem 1, the region $face_1$ is full-view covered by $\{C_1, C_2, C_3, C_4\}$. The camera sensor set $\{C_1, C_2, C_3, C_4\}$ is a minimal full-view coverage camera set, since if any camera sensor in $\{C_1, C_2, C_3, C_4\}$

is removed, $face_1$ will always be within the boundary of C_1C_3 or C_2C_4, i.e., it is impossible to full-view cover $face_1$.

3.1.4 The Advantages for Analytical Full-View Coverage Theory

The proposed analytical full-view coverage theory can be utilized to calculate the full-view covered area. If a region is judged to be full-view covered, then the coordinates of the boundary of the region can be determined by numerically solving Equation 3.7. Therefore, we can perform the numerical integration over the region and obtain its area. In Figure 3.4, the relationship between the full-view covered area and the sensing range is demonstrated, where each camera has the same effective angle $\theta_0 = \pi/3$ and all of them are uniformly distributed along a circle, resulting in that a corresponding region is full-view covered. Obviously, when we increase the sensing range of cameras, the area full-view covered region increases in a quadratic fashion.

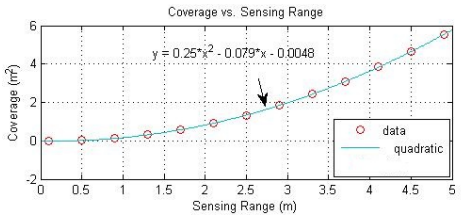

Figure 3.4: The relationship between the full-view covered area and the sensing range.

Theorem 2 *For a full-view covered simply connected domain, if the sensing range is increased by a factor k, then the full-view covered area can be increased by a factor of k^2, here k is any positive value.*

Theorem 2 is valid since we can construct a similar full-view covered simply connected domain if we increase the sensing range by a factor k. Because two similar figures that have the same shape though not necessarily the same size, the area of the latter figure will increase by a factor of k^2 Theorem 2 remains valid in the case of k-full-view coverage, which will be addressed in Section 3.2.

3.1.5 Basic Full-View Coverage Algorithm for a Point

In order to verify whether our algorithms or existing theories are correct or not, we first develop a preliminary algorithm that can be used to determine if a single point is full-view covered or not. This full-view coverage detection technique for a single point is important because it is the foundation for full-view covered area. If it can be proven that a single point in an area is not full-view covered, then the whole area is not full-view covered. Likewise, a region is full-view covered only if every point in it is full-view covered. By utilizing the same definition for full view coverage [1], we can conclude that any point will be full-view covered by a camera set (all elements in the set are within the sensing range to the

target) if the maximum angle θ_{max} between adjacent camera pairs is no greater than twice the effective angle, i.e., $\theta_{max} \leq 2\theta_0$.

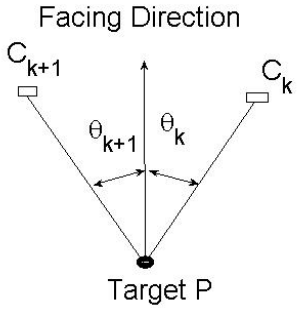

Figure 3.5: The angle between any adjacent camera pair should be no more than twice the effective angle, i.e., $\theta_{k+1} + \theta_k \leq 2\theta_0$

Suppose the angle between PC_{k+1} and PC_k is the maximum angle among all adjacent camera pairs, as shown in Figure 3.5. If $\theta_{k+1} + \theta_k > 2\theta_0$, then there exists a possible facing direction of the target that fulfills $\theta_{k+1} > \theta_0$ and $\theta_k > \theta_0$ simultaneously, meaning that no camera is sufficiently close to the target's facing direction. Thus, according to the definition of full-view coverage, the target is not full-view covered. On the other hand, if the maximum angle is less than or equal to twice the effective angle θ_0, no matter what direction the target faces, we have $\theta_{k+1} \leq \theta_0$ or $\theta_k \leq \theta_0$ or both. Hence, a fast algorithm for single point full-view coverage detection can be developed as follows:

Algorithm 1 Full-view coverage detection algorithm for a point P

Input: Camera set C within the sensing range R_s from the target.
Output: Determine whether a particular point P is full-view covered by C.

1: $C \Leftarrow$ all cameras within the sensing range R_s from the target.
2: Calculate all cameras' polar coordinates.
3: Sort all cameras' angles by the heap-sort algorithm.
4: Calculate the angle difference for every adjacent camera pair.
5: Find the maximum angle difference θ_{max}.
6: If $\theta_{max} \leq 2\theta_0$, the point P is full-view covered, otherwise is not full-view covered.

The following algorithm has the time complexity of $O(n \cdot \log(n))$, where n is the cardinality of the camera set C. This is because that we use the heap-sort algorithm to sort the angles of the cameras, and the operation times of the heap-sort dominate this algorithm. It is a fast algorithm because if every node stores its position information in polar coordinates, it can quickly determine whether or not it is within the sensing range R_s to the target P. A simple protocol establishes the desired camera set surrounding the target. By comparing each camera's angular component with its neighboring counterparts', the maximum angle among all adjacent camera pairs can quickly be found.

3.1.6 Full-View Coverage Hole Detection and Healing Solution

Coverage is considered as an important measure of the quality of service provided by camera sensor networks. In most applications involving full-view coverage, we are interested in reliable monitoring of the environment in such a way that there are no full-view coverage holes in the coverage. To achieve this goal, one possible methodology is to deploy camera sensors precisely according to a regular pattern such as an equilateral triangle. However, due to that the target field is hostile or unaccessible for human beings, it is difficult to make precise deployment realistic and random deployment can be a second choice. In this scenario, camera sensors may cluster in some place and leave full-view coverage holes in other places. On the other hand, even if we deploy camera sensor precisely at the beginning, a full-view coverage hole can still be generated as time goes on. Some camera sensors can deplete energy quickly or simply be destroyed by a hostile environment. Moreover, in energy-aware camera sensor networks, some camera nodes need to be powered off in order to save energy. Consequently, full-view coverage holes are created. Therefore, it is necessary to discover full-view coverage holes in order to ensure reliable monitoring. Once full-view coverage holes are determined, some mobile camera sensors can be utilized to patch the holes by making use of boundary information of these holes.

Full-View Coverage Holes in Randomly Deployed Camera Sensor Networks

Considering a collection of stationary camera sensors deployed randomly in a planar target field. Each camera sensor has the capabilities of local sensing and communication. We assume that each camera sensor has identical sensing range R_s and effective angle θ_0. Camera sensors are aware of their locations, and can obtain their neighbor's locations by exchange information with each other. Furthermore, we assume there are camera sensors located near the external boundary of the target field as illustrated in Figure 3.6, in order to eliminate the boundary effect in full-view coverage camera sensor networks, which are known as fence sensors and other camera sensors are referred to as internal camera sensors.

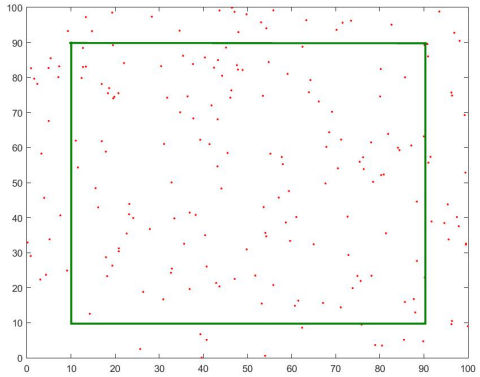

Figure 3.6: Full-view covered target area and its boundary region

In order to find all full-view coverage holes in the target field, we first find each camera sensor's full-view covered simply connected domain as shown in Figure 3.2. For a particular camera sensor c_0, we define it's full-view covered simply connected domain (FVCSCD) as follows:

Definition 4 *Camera sensor c_0's full-view covered simply connected domain D is an area that is full-view covered by c_0 and a subset of cameras C within the sensing range R_s from camera c_0 in such a way that if remove c_0, D will be not full-view covered by the subset of C again.*

After we find all FVCSCDs of c_0, we need to combine them since some of them are overlapped. We propose Algorithm 2 to obtain the boundary information of combined FVCSCDs for arbitrary camera sensor c_0 in the target field.

Algorithm 2 Combined Full-view covered simply connected domain detection algorithm for a single camera c_0

Input: camera c_0 and cameras set C_0 within the sensing range R_s from camera c_0.
Output: c_0's combined full-view covered simply connected domains.
1: $C_0 \Leftarrow$ all cameras within the sensing range R_s from the target.
2: find the possible Full-view covered simply connected domains of c_0.
3: combine overlapped portions of Full-view covered simply connected domains of c_0.
4: output boundary information of c_0's combined full-view covered simply connected domains.

For all camera sensors in the target field, we apply Algorithm 2 to obtain all corresponding combined FVCSCDs, which constitute the full-view covered area (indicated by A) in the target field. Obviously, the other area remained (indicated by A') in the target field is full-view covered holes. This statement can be proved by contradiction. Suppose any subregion in A' is not a hole, it must be full-view covered by a set of cameras in the target field, So there exists one particular camera (indicated by c_0) in this camera set, by definition 4, we can find a portion in A' (indicated by A''), which is a part of c_0's FVCSCD. So A'' is a part of A, this is a contradiction since A and A' are disjoint.

3.2 On k-Full-View-Coverage Algorithms

Many security and reliability applications require guaranteed k-coverage of an area at all times[55, 56, 57, 58, 59, 60, 61, 62, 63, 32]. As a result, k-coverage has attracted much attention in the past decade, but most of the previous work on k-coverage has focused on traditional scalar sensors instead of camera sensors. Camera sensors qualify as vector sensors because cameras from different positions can form various views of the target. The sensing quality of camera sensors depends on the angle between the facing direction of the target and the viewing direction of the camera. The concept of full-view coverage [1] is of great significance because it ensures both the detection and recognition of a target. A target experiences full-view coverage if, regardless of the direction it faces, it is always in view of a camera. At the same time, the camera's viewing direction should be sufficiently close to the direction the target faces

[1]. This research intends to investigate a special case of k-coverage called k-full-view-coverage. This case is where the target is in an over-deployed camera sensor network and k disjoint subsets of cameras are able to provide full-view coverage of the target. The two proposed algorithms aim to k-full-view cover both a single target and a given area based on equilateral triangle displacement.

3.2.1 Algorithms for Minimum Camera Set Problem

In this section, we discuss the minimum camera coverage algorithms. These algorithms aim to find the minimum camera set for given camera set. We first describe the problem and give its mathematical model. We will then present a greedy algorithm that is simple to implement. Finally, we develop a complex graphic algorithm and prove that it can achieve an optimum solution.

Given a camera set C where the Euclidean distance between each camera $c_i \in C$ and a point P is less than the sensing range R_s, i.e., $|Pc_i| \leq R_s$, find the minimum number of camera subsets $C_{min} \subseteq C$ that full-view cover P.

Let there be n cameras in the camera set C, where $c_i \in C$ has polar coordinates $\{r_i, \theta_i\}$, $r_i \leq R_s$ where R_s is the sensing range, and r_i is the distance from camera i to the target.

$$\forall i, j\{1 \leq i < j \leq n+1\},$$

$$\theta_i \leq \theta_j,$$

$$c_1 = c_{n+1},$$

This indicates that the camera set C has been sorted by the angular value in ascending order, using the heap sorting algorithm as mentioned in Algorithm 1.

Mathematically the problem can be formulated as: minimize $\sum x_i$ s.t.

1. $\sum_i^j x_k \leq 2, \forall i, j\{1 \leq i < j \leq n+1, mod(|\theta_j - \theta_i|, \pi) \leq 2\theta_0\}$.
2. $\sum_i^j x_k \geq 3, \forall i, j\{1 \leq i < j \leq n+1, mod(|\theta_j - \theta_i|, \pi) > 2\theta_0\}$.
3. $x_i \in \{0, 1\}$. $x_i = 1$ iff c_i is chosen, otherwise $x_i = 0$.

The first constraint meets the minimum requirement. The second constraint guarantees the full-view coverage condition. The third constraint interprets that $c_i = 0$ means c_i belongs to the minimum camera set and that $c_i = 1$ otherwise.

3.2.2 A Simple Greedy Algorithm for the Minimum Camera Set Problem

First we define the concept of a camera c_0's neighbor subset C as follows:

For any neighbor camera c_i in the subset C, in the counterclockwise direction, the angle θ between the neighbor c_i and c_0 is less than or equal to twice the effective angle θ_0, i.e., $\theta \leq 2\theta_0$.

We present the algorithm as follows:

Algorithm 3 Full-view coverage detection algorithm for a point P with minimum cameras

Input: Camera set C within the sensing range R_s from point P.
Output: Minimum subset of cameras that full view covers the point P.

1: $C \Leftarrow$ all cameras within the sensing range R_s from point P.
2: Sorting C by the angular value in ascending order by heap-sort algorithm.
3: $C' \Leftarrow \phi$.
4: Randomly select a camera from C and put it into C'.
5: Find its furthest neighbor counterclockwise between which the difference of angular values is less than or equal to twice the effective angle, put it into C'. If no such neighbor exists, set the minimum camera sbuset to empty, and go to step 7.
6: Iteratively repeat this process, until the last camera's neighbor set contains the first chosen camera.
7: Output the minimum camera subset.

For each camera selected, the angular difference between itself and it neighbor is not greater than twice the effective angle, so the sufficient condition for full-view coverage is satisfied. In each step, the algorithm always finds the furthest neighbor, counterclockwise, which is locally the best choice, so that the heuristic result can be achieved.

The algorithm will be terminated if the last camera is found when the first selected camera appears in its neighbor camera set, or if no such neighbor where the difference of angular values is less than or equal to twice the effective angle can be found. So the termination condition for algorithm is also satisfied.

For a selected camera, there are only two possibilities, one of which is that it finds a neighbor where the difference of angular values is less than or equal to twice the effective angle, or no such neighbor exists. So all cases are covered by the algorithm, which confirms the completeness of the algorithm.

The time complexity is $O(n \cdot \log(n))$.

Proof: For any camera chosen as the initial one in the subset, it will search its neighbors counterclockwise until finding its furthest neighbor. Then start from the furthest neighbor and repeat the same procedure until the first camera selected appears in the last chosen camera's neighbor subset. Essentially this is an iterative process with n comparisons. Hence, the time complexity is $O(n)$. Then we consider that the heap-sort algorithm is utilized to sort camera's angles, which has time-complexity of $O(n \cdot \log(n))$. Therefore the total time complexity is $O(n \cdot \log(n))$.

3.2.3 Optimum Algorithm for Minimum Camera Set Problem

We develop a graphic algorithm for the minimum camera set problem. It is similar to Dijkstra's Algorithm. We prove that it achieves an optimum solution.

First, for a given point P we use the technique described in Algorithm 1 to obtain the camera set where every element in the set has a distance from P of the sensing range R_s.

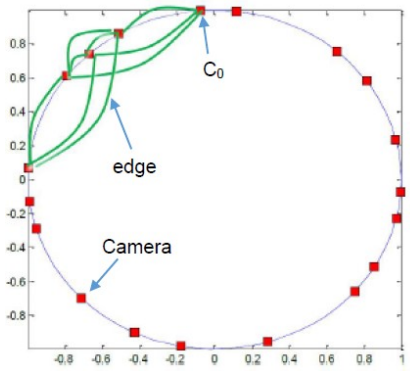

Figure 3.7: For every camera pair, if the angular difference is not greater than twice the effective angle, then add an edge between them.

Figure 3.8: A equivalent diagram from Figure 3.7.

Then place the cameras in a unit circle according to their angular position. For every arbitrary pair of cameras, if the angular difference is not greater than twice the effective angle, then add an edge between them, as shown in Figure 3.7.

1. Arbitrarily select a camera in Figure 3.7, e.g., C_0, as the starting node, and start from C_0. Straighten the circle illustrated in Figure 3.7 into a line as shown in Figure 3.8, where C_0 is both the starting node and ending node.

2. Perform Dijkstra's Algorithm on Figure 3.8 and find the shortest path (measured in hops) from the starting node to the ending node in the line.

3. For any other cameras in Figure 3.8, repeat steps 1 and 2 until there are no cameras left. Among all shortest paths we find the one with the minimum hops. This resultant shortest path is the minimum solution and the number of hops of this shortest path is the minimum number of cameras.

Proof: By performing Dijkstra's Algorithm for each camera we get the shortest path and the minimum camera number corresponding to the shortest path. In this sense, it is a local optimum solution. We obtain the global optimum solution by selecting the minimum camera number from all local solutions, hence, the final result is optimum. We can confirm this conclusion though contradiction. If a less optimum value exists then there exists a starting point in the shortest path starting from which we can implement the above graph algorithm. Hence from the same starting node and the same graph, we get two different minimum values, which is impossible.

Suppose the number of nodes is V (i.e., the number of cameras), the number of edge is E, then the time complexity $O(|V|\,(|E| + |V|\,log\,|V|))$.

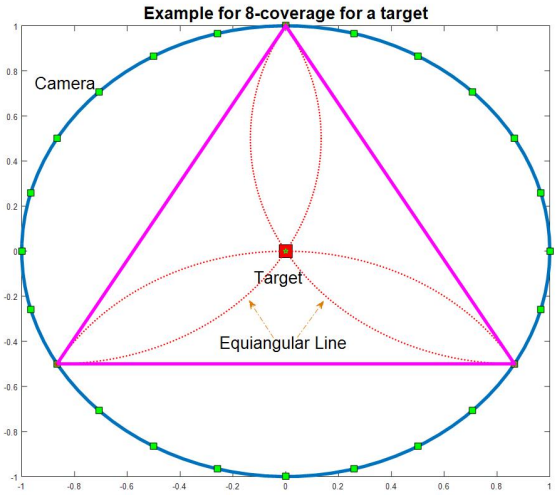

Figure 3.9: An example of 8-coverage for a target. On the circle, there are 24 uniformly distributed camera sensors which can be divided 8 disjoint camera subsets. Each subset can form an equilateral triangle and full-view cover the target.

Proof: For any camera that is chosen as a starting node, Dijkstra's Algorithm will be performed, which has a time complexity of $O(|E| + |V|\,log\,|V|)$ in the worst case. This process will be repeated V times, where V is equal to the number of cameras. Therefore the total time complexity should be $O(|V|\,(|E| + |V|\,log\,|V|))$.

3.2.4 k-Full-View-Coverage Theory and Algorithms

In this section we first propose the formal definition of k-full-view coverage, then we present the k-full-view coverage algorithm for a target (a point), and finally we propose the k-full-view coverage algorithm for a given area for equilateral triangle based displacement in camera sensor networks.

3.2.5 k-Full-View Coverage Concept

Definition 5 *k-full-view coverage for a target in camera sensor networks is when there exist k disjoint camera subsets, each of which can full-view cover the target.*

If any point in the given area can be k-full-view covered, then the area is k-full-view covered.

Figure 3.9 gives an example of 8-full-view coverage for a target. On the perimeter of the circle there are 24 uniformly displaced camera sensors surrounding the target, the effective angle for all cameras is $\pi/3$, and the sensing range is equal to the radius of the circle. If we divide these cameras into 8 disjoint camera subsets, where each camera subset constitutes an equilateral triangle as illustrated, then based on Algorithm 1, each camera subset will full-view cover the target. Since we have eight different disjoint camera sensor subsets, by the definition of k-full-view coverage, the target is 8-full-view-covered.

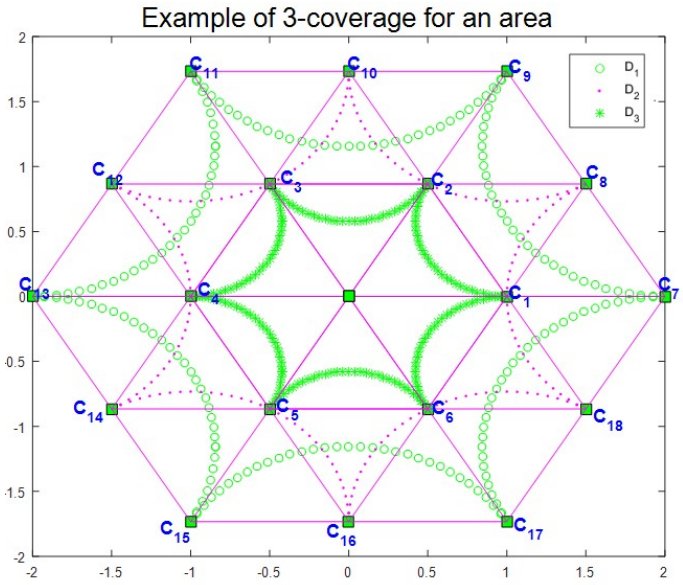

Figure 3.10: An example of 3-coverage for an equilateral triangular lattice based deployment area.

Figure 3.10 demonstrates an example of 3-full-view coverage for an area. Suppose the sensing range is 3 times the side length of the triangle and the effective angle is $\pi/3$. Then the six cameras on the vertices of the smaller inner hexagon will produce a single connected domain, indicated by D_1, where every point in this domain is full-view covered by the six cameras, through the application of Theorem 1. For the outer larger hexagon, we divide the 12 cameras located on the larger hexagon into two disjoint subsets, each of which will constitute a hexagon, and we generate a corresponding single connected domain, indicated by D_2, D_3. Since D_2, D_3 covers D_1, D_1 is 3-full-view covered by Definition 5.

3.2.6 k-Full-View-Coverage Algorithm for a Target

We propose in Algorithm 4, where given a camera set C and a target, that k disjoint camera subsets can be generated where each of k disjoint camera sets can full-view cover the point (target). We initially put all cameras into set C and let C' be the empty set. We also initialize DisjointSet to be empty and set Min to equal the number of camera sensors available from the input. Next, we sort the camera set C by the angular value in ascending order. The following steps use a for loop to produce a minimum camera subset that can full-view cover the target. This for loop is equivalent to Algorithm 3. After the for loop, we remove the produced minimum camera set C' from C, and let C' be one of the camera disjoint subsets. For the updated set C, we repeat steps 4-14 in order to create a new camera disjoint subset. If no more C' can be produced, the algorithm stops.

Lemma 2 *Without considering the boundary effect, for each equilateral triangle, if there exists a circle with the radius of $\sqrt{3}/3l$, centered at one of the vertexes of the triangle and the circle is full-view covered, then the whole area will be full-view covered.*

Algorithm 4 k-coverage algorithm for a point

Input: camera set C within the sensing range from target.
Output: k disjoint subsets, where each of them can full-view covers the point.

1: $C \Leftarrow$ all cameras within the sensing range from target.
2: $C' \Leftarrow \phi$.
3: $DisjointSet \Leftarrow \phi$.
4: $Min \Leftarrow n$.
5: sorting C by the angular value in ascending order.
6: **for** $i = 1$ to $C.length$ **do**
7: $temp \Leftarrow \phi$.
8: select the ith camera from C and put it into $temp$
9: find i's furthest neighbor between which the difference of angular values is less than or equal to twice the effective angle, put it into $temp$.
10: iteratively repeat this process, until the last camera's neighbor set contains the first camera.
11: calculate the number Num of the element in temp.
12: **if** $(Num < Min)$ **then**
13: Min=num,C'=temp.
14: **end if**
15: **end for**
16: remove C' from C.
17: $DisjointSet \Leftarrow C'$.
18: repeat step 4-14 until C' is ϕ.
19: output DisjointSet.

Proof: It is well known that to cover an equilateral triangle with circles of the same and smallest size, the minimum number of circles should be 3, and each circle's diameter is the line between the vertex and the center of the triangle, as shown the smaller circles in Figure 3.12. Hence, if each circle is centered on one of the vertexes of the triangle and passes the center of the triangle as shown the bigger circles in Figure 3.12, these three circles will cover the equilateral triangle too.

3.2.7 k-Full-View-Coverage Algorithm for an Area

Since the triangular lattice deployment is the best choice to cover a given area with the minimum number of sensors, we explore the k-full-view coverage algorithm for equilateral triangular deployment.

Theorem 3 *Without considering the boundary effect, if the circle that is centered at one of the vertices of the triangle and has a radius that is* $\sqrt{3}/6$ *of the side length of the triangle is k-full-view covered, then the entire given area is full-view covered.*

Proof: It is well known that in order to cover an equilateral triangle by 3 identical circles, the minimum radius of the circle should be equal to $\sqrt{3}/6$ of the side length of the triangle, and the three circles should be located on the three vertices of the triangle. If one of the circles is k-full-view covered, the other two circles must also be k-full-view covered, due to the isotropy of the triangular lattice. For the same reason, if one of the triangles is k-full-view covered, then all of the triangles should be k-full-view covered, if the boundary effect is ignored.

Definition 6 *The circle that is centered at one of the vertices of the triangle and has a radius that is*

$\sqrt{3}/6$ *of the side length of the triangle, is called Basic Circle (BC).*

Algorithm 5 k-coverage algorithm for triangle lattice based deployment area

Input: camera sensors deployed in equilateral lattice, the BC centered at O, sensing range R_s and effective angle θ_0.

Output: k disjoint subsets of cameras where each of them can full-view cover the BC.

1: $C' \Leftarrow \phi$.
2: Using O as a center, find the least number of vertices of regular polygon P_{least} constituted by camera sensors whose corresponding single connected domain formed by equiangular lines contains BC and whose diagonal is greater than or equal to R_s.
3: If no such P_{least} exists, go to step 7;
4: $C' \Leftarrow P_{least}$.
5: remove cameras inside P_{least} from camera lattice;
6: repeat steps 2-5.
7: output C'

In Algorithm 5, we ignore the boundary effect of full-view coverage. P_{least} also stands for the camera set that constitutes the corresponding regular polygon.

3.3 On Camera Sensor Density Minimization Problem for Triangular Lattice-Based Deployment

It has been proven that triangular deployment is the best choice to cover a given area with minimum sensors[40]. In this paper, we investigate in detail the camera sensor density minimization problem for triangular lattice based deployment in full-view covered camera sensor networks. We prove the formula in literature [1] which aims to minimize triangular lattice density is not correct. We then demonstrate a minimization formula for equilateral triangle lattice-based deployment and prove its correctness. Finally, we perform simulations and the results show our conclusions are correct.

3.3.1 The Flaws for Camera Sensor Density Minimization Problem for Triangular Lattice- Based Deployment

Based on Algorithm 8, we can prove Lemma 5.1 in Wang *et al* [1] is not correct. In Lemma 5.1, a necessary and sufficient condition on the sensor density for full-view coverage in a triangular lattice based deployment is given. We rewrite the Lemma as follows:

$$l = l(R_s, \theta_0) = \frac{2R_s}{\sqrt{3} + cot\theta_0}$$

Here l, θ_0 and R_s are the side length of the equilateral triangular, the effective angle and the sensing range respectively. Hence the ratio of sensing range R_s over the side length l is

$$R_s/l = \sqrt{3}/2 + cot\theta_0/2$$

Figure 3.11 (a) shows the relationship between ratio R_s/l and effective angle θ_0. When effective angle θ_0 is between 75° and 90°, i.e., $75° \leq \theta_0 < 90°$, the sensing range R_s will be less than the side length l of the equilateral triangle. This implies the position of each camera will be not within the sensing range R_s of any other cameras, so if the target in the position of any camera, it will not be full-view covered. Therefore, we can conclude that the statement in Lemma 5.1 in [1] which says any point in the area is guaranteed to be full-view covered is not correct.

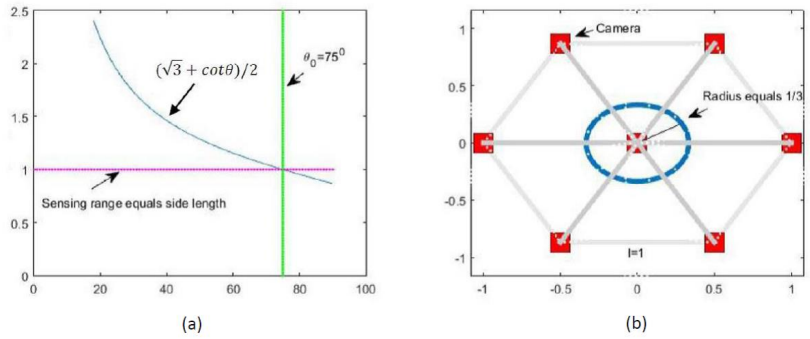

Figure 3.11: (a) When the effective angle $75° \leq \theta_0 < 90°$, the sensing range R_s will be less than the side length of the equilateral triangle, resulting in the position where each camera is located will be out of reach of any other camera sensors, hence not being full-view covered and contradicting the statement in Lemma 5.1 in [1]. (b) Given effective angle θ_0= 75°, simulations show 10000 points randomly generated in the area bounded by the circle of which the radius equals one-third of side length l will not be full-view covered.

In Figure 3.11 (b), the effective angle θ_0 equals 75°, the side length l of triangle equals one unit, and the sensing range is obtained by applying Lemma 5.1 from Wang *et al*'s paper [1]. The circle is centered at the origin and with a radius of one-third of the side length of the equilateral triangle. Using Algorithm 1, we performed simulations to further validate the above observation by randomly selecting 10000 points which are located inside the sub-area bounded by the circle. The results show that none of the points are full-view covered, which contradicts Lemma 5.1 of Wang *et al*'s paper [1]. Hence, we conclude that there exist defects in the full-view coverage theory in [1] and propose novel theories to solve the problems in equilateral triangle lattice based deployment which will be illustrated in Section 3.3.2.

3.3.2 Minimization for Equilateral Triangle Lattice Based Deployment

It has been proven that triangular lattice deployment is the best choice to cover a given area with minimum sensors. Figure 3.14 illustrates this statement. In this section, we demonstrate a minimization formula for equilateral triangle lattice-based deployments and show its correctness.

Density minimization for Triangle Lattice-Based Deployment

The minimum side length is given as follows $\frac{R_s}{k+\sqrt{3}/3}$, here k depends on the effective angle and k=1,2,3,...

if effective angle $\pi/2 > \theta \geq \pi/3$, then k=1;

if effective angle $\pi/3 > \theta \geq 0.38$, then k=2;

if effective angle $0.38 > \theta \geq 0.21$, then k=3; and so on;

Before we prove above conclusion, we present a lemma as follows:

Lemma 3 *for each equilateral triangle, if there exists a circle with a radius of $\sqrt{3}/3l$, centered at one of the vertexes of the triangle and the circle is full-view covered, then the whole area will be full-view covered.*

Proof: It is well known that to cover an equilateral triangle with circles of the same and smallest size , the minimum number of circles should be 3, and each circle's diameter is the line between the vertex and the center of the triangle, as shown the smaller circles in Figure 3.12. Hence, if each circle is centered on one of the vertexes of the triangle and passes the center of the triangle as shown the bigger circles in Figure 3.12, these three circles will cover the equilateral triangle too. Due to the isotropy of

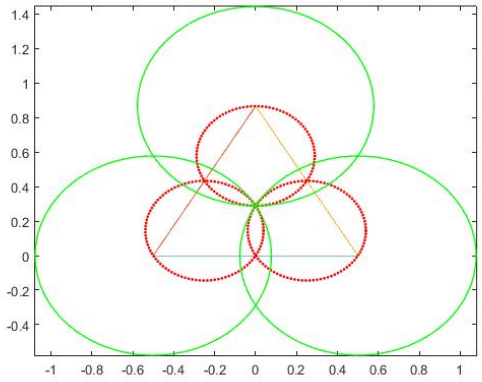

Figure 3.12: The smaller circles have the same and smallest size to cover the equilateral triangle. The bigger circle is centered at one vertex of the triangle and passes the center of the triangle. The three bigger circles will cover the triangle too. If one of the bigger circles is full-view covered, then the whole area is full-view covered due to the isotropy of equilateral triangle deployment.

equilateral triangle deployment, if one of the bigger circles in Figure 3.12 is full-view covered, then the other two should be full-view covered too. Since the distance from vertex to the center of the triangle equals $\sqrt{3}/3l$, so the circle has a radius of $\sqrt{3}/3l$.

The full-view covered area looks like this in the following figure.

Disk covering: Hexagonal covering has an optimal density (Kershner, 1939). Covering density is $\frac{2\pi}{3\sqrt{3}}$

Sphere covering: cover infinite space with identical spheres as "thinly" as possible. The quality measured by covering density: vol(spheres)/vol(enclosing sphere)

If the effective angle is large enough, we need only the first ring (defined as the hexagon in 3.13) constituted by 6 camera sensors to full-view cover the circle, as shown in Figure 3.13. In Figure 3.13, the circle is inscribed with all equiangular lines (an equiangular line is a set of points where the angle

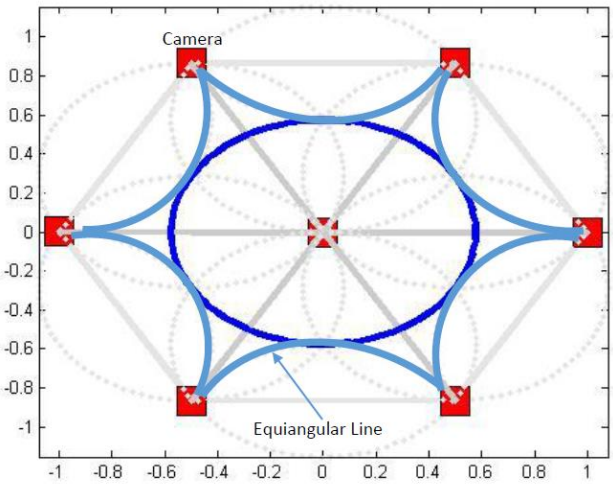

Figure 3.13: The area bounded by the circle is full-view covered by 6 cameras in the first ring where the effective angle $\pi/2 > \theta \geq \pi/3$. The circle is centered at one vertex of the triangle and passes the center of the triangle. The circle is inscribed with the boundary consisting of equiangular lines. if the circle is full-view covered, based on Lemma 3, the whole target field will be full-view covered.

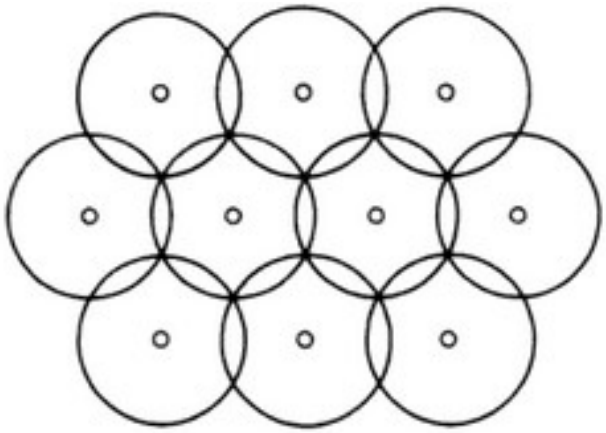

Figure 3.14: Covering the plane with circles, where the centers belong to the hexagonal lattice, which is the most efficient covering since there is least overlap among the circles.

between two adjacent camera sensors is identical). After simple calculation, we can get the lower bound of the effective angle is $\pi/3$. When the effective angle increases, the full-view covered area bounded by the equiangular lines will enlarge too, which guarantees the circle is full-view covered. The side length is minimum because if we increase the minimum side length, given the sensing range and the scope of the effective angle, The area bounded by the equiangular lines will overlap with the circle shown in Figure 3.13, which means the circle no longer full-view covered. In order to cover the whole circle in Figure 3.13, the sensing range $R_s = (1 + \sqrt{3}/3)l$, so if effective angle $\pi/2 > \theta \geq \pi/3$, then the minimum side length equals $\frac{R_s}{k+\sqrt{3}/3}$, where k=1.

If the effective angle $\pi/3 > \theta \geq 0.38$, the area enclosed by the equiangular lines will inscribe with the circle in Figure 3.15. The circle has a radius of $\sqrt{3}/3l$ and centered at the origin. In this scenario, the twelve cameras in the second ring (defined as the middle hexagon in 3.15 constituted by 12 camera

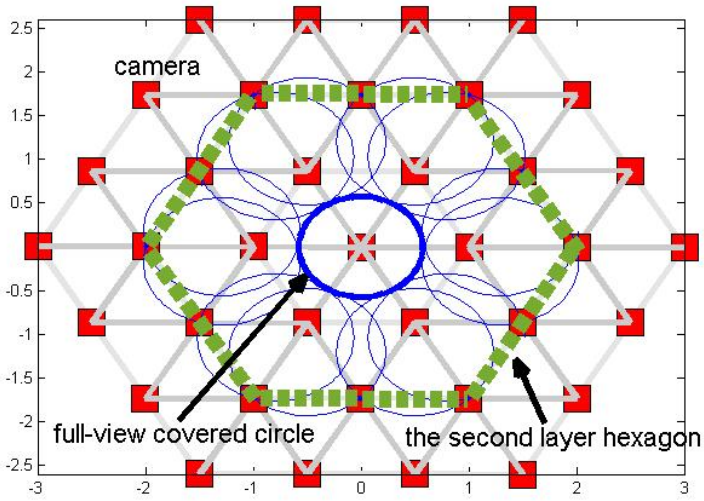

Figure 3.15: The area bounded by the circle is full-view covered by 12 cameras in the second ring where the effective angle $\pi/3 > \theta \geq 0.38$.

sensors, shown by the doted lines) should participate the full-view coverage process. The approximate critical effective angle of 0.38 is obtained by the numerical calculation methods. For the 12 cameras in the second ring, we can draw the corresponding boundary enclosed by the equiangular lines (only the arcs within the second ring are equiangular lines). The equiangular boundary will shrink with the decrease of the effective angle. When the effective angle decreases to 0.38, the equiangular boundary will inscribe with the circle as shown in Figure 3.15. Hence we prove if effective angle $\pi/3 > \theta \geq 0.38$. The minimum side length equals $\frac{r}{k+\sqrt{3}/3}$, where $k = 2$; The side length is minimum for the same reason illustrated in Figure 3.15.

Similarly, if the effective angle continues to decrease from 0.38, the 18 cameras in the third ring (defined as the outer hexagon in figure 3.15 constituted by 18 camera sensors) must participate in order to full-view cover the circle as shown in Figure 3.15. we can get the minimum side length for this scenario. The analysis is similar so we omit it for simplicity. And so on for even small effective angles.

We perform simulations to verify the correctness of our formula. The results show that the target field is full-view covered under the deployment following our formula.

In conclusion, we prove one important formula in previous literature is not correct, and propose our solution and conduct simulations to justify its correctness.

Chapter 4

Energy Issues in Full-view Coverage

The available energy is an important bottleneck in the operation of camera sensors since camera sensors usually depend on their battery for power supply. Therefore, the energy issues should be considered as a major constraint while designing protocols in full-view coverage camera sensor networks. Particularly, replenishment of power resources might become impossible in most application scenarios, resulting in that the lifetime of camera sensor network depending on battery lifetime. In this chapter, we address the lifetime and full-view coverage guarantees through distributed algorithms in camera sensor networks and object tracking problems.

4.1 Lifetime and Full-View Coverage Guarantees Through Distributed Algorithms in Camera Sensor Networks

When a vast number of camera sensors are randomly placed in a target field, how can we produce a sleep/activate schedule for camera sensors to maximize the lifetime of target coverage in the target field? This well-known problem is called Maximum Lifetime Coverage Problem (MLCP)[64, 61, 29], which has been explored extensively in the literature. However, MLCP in full-view coverage camera sensor networks are far more complicated since each target should be full-view covered by a group of camera sensors instead of a single sensor as in conventional scenarios.

In this section we address the full-view covered target problem, with the objective of maximizing the lifetime of a power constrained camera sensor network deployed for full-view coverage of targets with a known location. We consider that a large number of camera sensors are placed randomly in close proximity of targets and each camera sensor sends the collected data to a base station. We define the camera sensor network lifetime as the time interval the targets are full-view covered by a camera sensor cover.

Our contributions include: (1) an efficient data structure to represent a family of camera sensor

cover, each of which full-view cover all targets with consideration of the lifetime of each camera sensor; (2) an energy efficient monitoring scheme where only the camera sensors from the current active set are responsible for monitoring all targets and for transmitting the collected data, while all other camera sensors are in a low-energy sleep mode; (3) formulating the maximizing camera sensor network lifetime problem into a packing linear programming problem, where a modified Garg-Konemann Algorithm to find the near optimal solution is applied.

4.1.1 Maximum Lifetime Problem for a Full-View Covered Target

Network Model

In this section, we will study reducing power consumption rate of camera sensors in coverage for a single target P by alternating the sensor modes in order to prolong the full-view coverage lifetime for the target. The target lifetime is defined as the time interval from the activation of full-view coverage of the target until the first time at which the target is no longer possibly full-view covered. We assume that the target is within the sensing range R_s of a group of camera sensors C, where C is a set of camera sensors that are also called nodes, and the cardinality $|C| = n$. Each node $c \in C$ can monitor events of interest in its sensing range R_s and communicate with nodes in its transmission range R_t, where $R_t > R_s$.

In order to facilitate communication with each other for $c \in C$, we assume each camera sensor $c \in C$ has a unique identification number indicated by $ID(c)$. We assume that nodes have distance information about $c \in C$ and P represented by polar coordinates. Specifically, each node $u \in C$ knows the polar coordinate (r_v, α_v) of $v \in C$. We assume the polar coordinate of the target P is $(0, 0)$. Let $N(u, v) \mid \alpha_u \leq \alpha_v$ denotes the number of nodes w that satisfies $\alpha_u < \alpha_w < \alpha_v$. If $N(u, v) = 0$, then u and v are defined as *neighbors* of each other.

Let $b_c(0)$ be the initial energy of camera sensor $c \in C$. As a normalization, we assume that each camera sensor consumes 1 unit of energy in each time slot if it is active. If it is in sleep mode, then it consumes the energy of zero. Let $\hat{t} = max_{u \in C} b_u(0)$. let Υ stands for the node corresponding to $\hat{t}$, where if there are multiple nodes have the same maximum initial energy, randomly select one as Υ. We divide the time into *time slots* and each camera sensor is equipped with synchronized clocks which notify them at the beginning of each time slot.

An Efficient Data Structure

To full-view cover the target P, we consider a unit circle which is divided into $\lceil 2\pi/\theta_0 \rceil$ sections as shown in Figure 4.1. Since the target is within the sensing range R_s of all camera sensors, only the angular position α_c of $c \in C$ relative to target P affects the full-view coverage for the target, the Euclidean distance between $c \in C$ and the target does not have influence on the full-view coverage for the target.

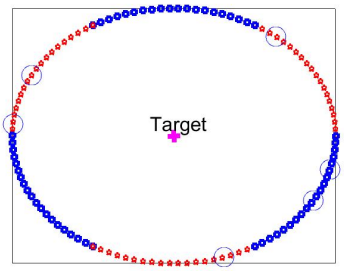

Figure 4.1: An example of equivalent unit circle on which camera sensors are distributed

.

If we move $c \in C$ from its original position to the unit circle radially, while maintaining its same angular position, we will achieve the same full-view coverage for the target with equivalent constraints.

At the initialization phase, each camera sensor exchanges and collects information with its neighbors such as ID, coordinates, energy etc. The process starts from a randomly selected camera sensor, which sends a message to its counter clockwise neighbor. This neighbor will attach its information to the message list, and then forward the message to its counter clockwise neighbor. This process continues until the information of all camera sensors are collected and received by the first camera sensor, which will forward the message to its neighbor until all camera sensors share it. Eventually, each camera sensor will obtain the information of all others.

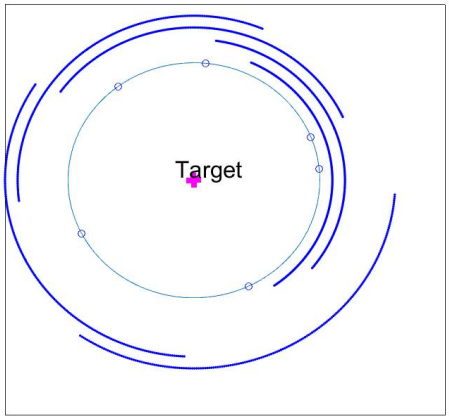

Figure 4.2: Partial-views provided by camera sensors

Given the angular position of each camera sensor $c \in C$, we define the partial view $c \in C$ as the angular interval $(\alpha_c - \theta_0, \alpha_c + \theta_0]$, where this specific angular interval of the target can be monitored by $c \in C$. As shown in Figure 5.2, there are six camera sensors on the unit circle, each one has a corresponding arc denoting the partial view range, whose angular diameter from the target equals to $2\theta_0$.

For each camera sensor $c \in C$, we define a piecewise function:

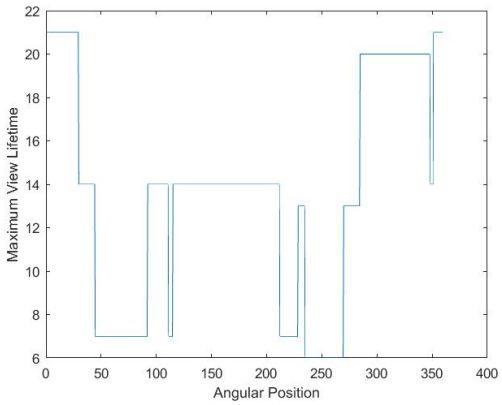

Figure 4.3: An example of $\sum_{c \in C} f(\theta_c)$

$$f(\theta_c) = \begin{cases} b_c, & \alpha_c - \theta_0 \leq \theta \leq \alpha_c - \theta_0 \\ 0, & otherwise \end{cases}$$

Each piecewise function represents the maximum lifetime to partially-view cover the angular range $\alpha_c - \theta_0 \leq \theta \leq \alpha_c + \theta_0$ by $c \in C$.

We then sum up the piecewise functions of all camera sensors and get

$\sum_{c \in C} f(\theta_c)$.

After collecting information from neighbor camera sensors, matching nodes to the unit circle, determining the partial view interval and generating the piecewise function for each camera sensor, we obtain a function $\sum_{c \in C} f(\theta_c)$ as shown in Figure 4.3, the horizontal axis stands for the angular range $[0^o, 360^o]$, the vertical axis stands for the maximum view lifetime, which is defined as the maximum time of partial view coverage for a particular angular location. This function is equivalent to a very efficient data structure since it can not only show whether a target could be full-view covered or not but also shows the maximum monitoring lifetime for the target. For example, in Figure 4.3, the target will be full-view covered for at most 6 units of time, which is the minimum of maximum view lifetime in the interval $[0^o, 360^o]$. After that, the target will not be full-view covered, unless the angular interval $[230^o, 270^o]$ is partially view covered by extra camera sensors. This data structure also shows the camera sensors with residual energy after the maximum life of full-view coverage. In the multiple targets scenarios, these camera sensors with residual energy can be utilized to full-view cover other targets, which we will discuss in the next section.

Distributed Algorithm for Maximization lifetime on Full-View Covering a Target

We now describe the Distributed Camera Sensor Cover (DCSC) algorithm that we propose. After obtaining the information of all camera sensors, each camera node first sort all angular position values in ascending order. Then generate all piecewise function and sum up to get the function $\sum_{c \in C} f(\theta_c)$, where the minimum value of maximum view lifetime can be found, along with the corresponding camera

sensors. We check the minimum value is nonzero or not, if not, we reach the conclusion that the target is impossible to be full-view covered. Otherwise, starting from one of these camera sensors that have minimum value b_{min} of maximum view lifetime, we pick it into one of camera sensor covers denoted by s. Then we find the next camera sensor counterclockwise which least overlaps the previous one. We repeat this procedure until all angular positions are view covered. This camera sensor cover s is a qualified one that can guarantee full-view coverage and maximizing the lifetime for the target, which is put into sensor cover set S. Next, we reduce b_{min} by one unit, and repeat the previous procedure and get another sensor cover which is also put into S. Repeat this process until $b_{min} = 0$. The resultant camera sensor cover set S is the one that can maximize lifetime on full-view covering a target.

Algorithm 6 The Distributed Camera Sensor Cover (DCSC) algorithm

Input: A lifetime vector $b \in R^m$, an effective angle θ_0 and an angular positions vector α.
Output: full-view coverage camera sensor cover set.

1: sort α in ascending order;
2: generate $\sum_{c \in C} f(\theta_c)$;
3: find the minimum b_{max} of maximum view lifetime.
4: $S_0 \leftarrow \emptyset$
5: $s \leftarrow \emptyset$
6: while $b_{max} > 0$:
   $b_{max} \leftarrow b_{max} - 1$;
   Find the camera sensor c falling in the angular interval corresponding the minimum of b_{max};
   $s \leftarrow c$;
   Counterclockwisely find the camera sensor c' that least overlaps with c;
   $s \leftarrow c'$;
   Repeat the above step until all angular position is partially view covered.
   $S_0 \leftarrow s$;
7: Output S_0.

4.1.2 Maximum Full-View Coverage Lifetime Problem for Multiple Targets

Motivation

Consider a group of camera sensors which are randomly deployed surrounding multiple targets. What is the better quality of service (surveillance) that the camera sensors can provide? In the case of full-view coverage, a lot of literature focus on the minimization of camera sensors to full-view cover a target field. However, in the perspective of energy, minimizing the number of camera sensors is not the better solution. As mentioned in the previous section, after depletion of the energy of some critical camera sensors which lead to full-view cover a target, there are some camera sensors left with high residual energy which can be utilized to full-view cover other targets or by introducing extra camera sensors to renew full-view coverage for the current target.

Voronoi Diagram

When we consider multiple targets and a group of camera sensors, how to distribute these camera sensors among the targets? Intuitively, the closer the camera sensor to the target, the greater the possibility the target is within the camera sensor. Hence, we introduce the Voronoi Diagram to partition the camera sensors among targets.

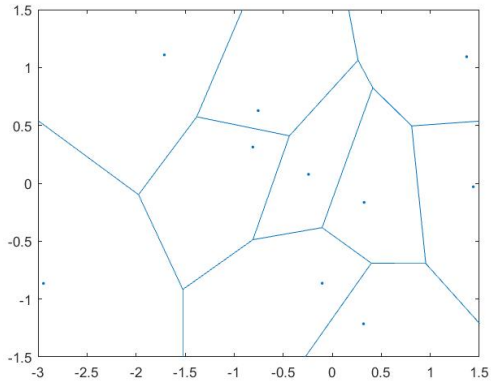

Figure 4.4: In the first selection process, the camera sensors are divided into groups based on Voronoi Diagram, where camera sensors which are closer to the target are grouped together.

As shown in Figure 4.4, each point in one of the polygons stands for a target. All camera sensors in the polygon are responsible for monitoring this target, unless specifically designated. Next, we group all camera sensors together that make the target within the sensing range of camera sensors, and execute Algorithm 6. This execution will lead to camera sensors with residual energy left, and we develop a new algorithm to further maximize the lifetime of full-view coverage for multiple targets.

Problem Statement

Given a group of camera sensors mentioned in the above subsection surrounding multiple targets, what is the longest lifetime for the targets to be full-view covered? Our goal is to produce a family of sensor covers $C_1, ..., C_m$, each of which can full-view cover the targets. Note the sensor covers are not disjoint. Then we setup a monitoring schedule which constitutes pairs $(C_1, t_1), ..., (C_m, t_m)$, where C_i is a camera sensor cover and t_i is time during which C_i is active. The mathematical problem then arises: how to determine $C_1, ..., C_m$ to maximize $\sum_{i=1}^{m} t_i$? We provide the formal definition of energy efficient full-view coverage problem in this subsection.

Maximum Full-View Coverage Lifetime Problem. Given a set of targets P, a set of camera sensors $c_1, ..., c_n$ and residual energy supply b_i for each camera sensor, find a monitoring schedule $(C_1, t_1), ..., (C_m, t_m)$ with the maximum length $t_1 + ... + t_k$, such that for any camera sensor c_i the total active time does not exceed b_i.

We can create a matrix $(C_{ij}) \in \Re^{m \times n}$, where rows $i = 1, ..., m$ represents each camera sensor and

columns $j = 1, ..., m$ represents each sensor cover and

$$C_{ij} = \begin{cases} 0 & \text{if sensor i is not in sensor cover j} \\ 1 & \text{if sensor i is in sensor cover j} \end{cases}$$

We get the packing linear programming[65, 66, 67, 68, 69] as follows:

maximize: $\sum_{j=1}^{m} t_j$

subject to

1. $\sum_{j=1}^{m} C_{ij} t_j \leq b_i$.
2. $\sum_{i=s}^{t} C_{ij} \leq 2,\ \forall s, t\{1 \leq s < t \leq n+1,\ \mod(|\theta_s - \theta_t|, \pi) \leq 2\theta_0\}$.
3. $\sum_{i=s}^{t} C_{ij} \geq 3,\ \forall s, t\{1 \leq s < t \leq n+1,\ \mod(|\theta_s - \theta_t|, \pi) > 2\theta_0\}$

In the objective function, the optimization problem is to find an assignment of values to t_j that satisfies the collection of the first linear inequalities, where the sum of active time in each camera sensor cover for an individual camera sensor in all camera sensor covers should be less than its lifetime of b_j. For each feasible solution, it should also satisfy the second and third inequalities. In the second collection of inequalities, if the angle between any two camera sensors with respect to the target is less than or equal to twice the effective angle θ_0, then the number of camera sensors should be less than or equal to two, otherwise, if we take out the camera sensor in between and the target will still be full-view covered.

Application the modified Garg-Könemann Algorithm

We apply the well-studied Garg-Könemann algorithm[70, 71, 72, 73, 74] for solving packing linear programs, which can solve the Maximum Sensor Network Life problem with nearly the same guarantee with its dual problem such as Minimum Weight Sensor Cover Problem. In Algorithm 7, each camera sensor is assigned a weight that is an exponentially increasing function of the energy it has consumed so far. The camera sensor cover with minimum total weight among all those that full-view covers all targets will be activated. This selection mechanism balances the monitoring load on all the camera sensors, and preferentially selects in each slot the sensors with high residual energy.

Implementation of Algorithm 7

To implement Algorithm 7, we need to collect the information of all camera sensors. To achieve this goal, we first assign the camera sensor with most abundant residual energy as cluster leader among all camera sensors that surround one particular target. We assume the communication range is great enough so that the cluster leader can communicate with its counterpart of its neighbor target. Hence, each cluster leader can collect the energy information of all camera sensors and perform Algorithm 7. By applying Algorithm 7 to solve the LP to maximize $\sum_{j=1}^{m} t_j$ under constraint $\sum_{j=1}^{m} C_{ij} t_j \leq b_i$, we get the required sensor covers to meet the energy constraint. We then check all solutions under constraints

$\sum_{i=s}^{t} C_{ij} \leq 2,\ \forall s,t\{1 \leq s < t \leq n+1,\ \mod(|\theta_s - \theta_t|, \pi) \leq 2\theta_0\}$ and $\sum_{i=s}^{t} C_{ij} \geq 3,\ \forall s,t\{1 \leq s < t \leq n+1,\ \mod(|\theta_s - \theta_t|, \pi) > 2\theta_0\}$. If one solution satisfies these constraints, we can guarantee the sensor cover can full-view cover the targets, otherwise, it fails to do so and remove it. Finally, we get all sensor covers, among of which each can full-view cover the targets.

Algorithm 7 The modified Garg-Könemann Algorithm with f-approximate minimum length columns

Input: A vector $b \in R^m, \epsilon > 0$, and an f-approximation algorithm F for the problem of finding the minimum length column $A_{q(y)}$ of a packing LP $\{max\ c^T x \mid Ax \leq b, x \geq 0\}$

Output: A set of columns $\{A^j\}_{j=1}^k$ each supplied with the value of the corresponding variable x^j, such that $(x^1, ..., x^k)$ correspond to all non-zero variables in a near-optimal feasible solution of the packing LP $\{max\ c^T x \mid Ax \leq b, x \geq 0\}$.

1: Initialize: $\delta = (1+\epsilon)((1+\epsilon)m)^{-1/\epsilon}$, for i=1,...,m, $y(i) \leftarrow \frac{\delta}{b(i)}$, $D \leftarrow m\delta$, j=0.

2: while $D < 1$
Find the column A_q using the f-approximation F.
Compute p, the index of the row with the minimum $\frac{b(i)}{A_q(i)}$ $j \leftarrow j+1$, $x^j \leftarrow \frac{b(p)}{A_q(i)}$, $A^j \leftarrow A_q$ For i=1,...,m, $y(i) \leftarrow y(i)(1 + \epsilon \frac{b(p)}{A_q(p)/\frac{b(i)}{A_q(i)}})$, $D \leftarrow b^T y$.

3: Output $\{(A^j, \frac{x^j}{log_{1+\epsilon}^{\sigma}})\}_{j=1}^k$.

Simulations and Evaluation

In the simulations, we assume that the camera sensors are deployed randomly and each camera sensor node has 10 units of initial energy. The camera sensor area is 1000*1000 m^2. The monitored area is 800*800 m^2. The number of targets in the monitored area is 9. In Algorithm 7, $\epsilon = 0.5$. The locations of each camera sensor and target are fixed, the effective angle $\theta_0 = \pi/4$, while the sensing range are changeable and are the same for all camera sensors in each simulation.

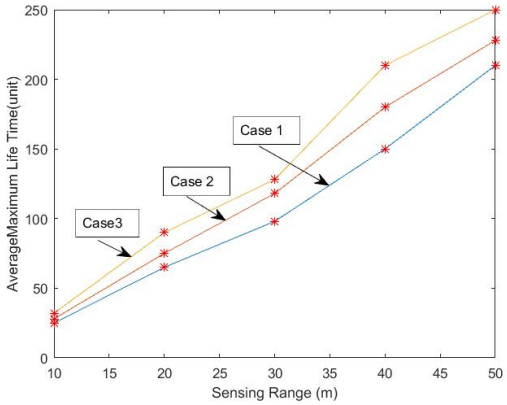

Figure 4.5: The average Maximum Lifetime vs the Sensing Range.

We consider three cases for simulations. In case one, none of the targets can share camera sensors with its neighbors since the targets are separated sufficiently further and none of camera sensors can monitor more than one target. This case is equivalent to the maximum lifetime problem for a single target. In case two, we assume a target could share a group of camera sensors that is sufficient to achieve

1-full-view coverage for both targets. Compared with case 1, the targets are more closely deployed. In case three, we assume that any one target can share three disjoint groups of camera sensors with its neighbors, each group of camera sensors can full-view cover the target itself and one of its neighbors.

We perform simulations 50 times for each case and compute the average maximum lifetime for the corresponding sensing range. In Figure 4.5, the relationship between the average maximum lifetime of camera sensor network and the sensing range of camera sensor is demonstrated. With the increase of the sensing range, the number of camera sensor available for k-full-view coverage is increased rapidly, resulting in the increase of average maximum lifetime. On the other hand, with the increase of sensing range, the power required to monitor the target will increase correspondingly. Therefore, the average maximum lifetime does not increase dramatically. In case two, the average maximum lifetime is greater that in case one, since any target can utilize more potential residual energy from the camera sensors which are supposed to dedicate to its neighboring target. In case three, we can observe that the average maximum lifetime are greater than that in case two, since more potential residual energy from the camera sensors dedicating to its neighboring targets can be exploited.

4.1.3 Conclusions

In this section, we tackle the challenge to maximize the full-view coverage for a target or multiple targets. We develop a distributed algorithm to find camera sensor cover set, where each of them is activated while other camera sensors remain asleep. We develop an efficient data structure which not only shows a target is full-view covered or not but also shows how long at most this target is full-view covered. We then apply Garg-Konemann algorithm to achieve maximization of lifetime of multiple targets while guaranteeing full-view coverage.

4.2 Object Tracking Problems in Full-View Coverage Camera Sensor Networks

Object tracking in camera sensor network is a quite representative application which has widespread use in applications such as security surveillance and wildlife habitat monitoring [75, 76, 30, 77, 78, 79]. During tracking period, we expect that camera sensors remain largely inactive for long periods of time, but then becoming suddenly active when the moving object which we need to monitor is detected. Tracking involves a significant amount of collaboration between individual camera sensors to perform full-view coverage. This collaborative nature of camera sensor networks offers significant opportunities for energy management. For example, just the decision of whether to do the collaborative full-view coverage at the user end-point or somewhere inside the network has significant implication on energy and lifetime. We focus on the issue of minimization of the power consumed by camera sensor nodes,

which involves the object tracking. During the object tracking, some camera sensors located alongside the object moving path will be turned on to achieve full-view coverage goal at the beginning of tracking object and turned off to save energy after the object moves away, the timing of turning on/off a particular camera sensor and the switch of adjacent camera sensors depend on the object's movement and location information. So, in this case, the camera sensor network is not a continuous-connected one. It is designed for intermittent network connectivity. We will develop a brute-force algorithm and a heuristic algorithm to achieve energy optimization goal during the object tracking process in the full-view coverage camera sensor networks.

4.2.1 Problem Statement

During the tracking period, the camera sensors remain largely inactive for longer periods of time. When the moving objects we need to monitor are detected, they become suddenly active. The problems are: How to collaborate individual camera sensors to perform full-view coverage? Which ones will be activated? When will they be activated? How long will they be activated?

Almost all existing tracking algorithms assume that the object motion is smooth with no abrupt changes – The object motion is assumed to be of constant velocity. We suppose prior knowledge about the initial position of the target $\vec{r_0}$ and its velocity $v(\vec{t})$, i.e.,

$$r(\vec{t}) = \vec{r_0} + \int_0^t v(\vec{t})dt$$

By broadcasting, we assume the target's position, speed, and moving direction are know for all camera sensors. As a consequence, the path of the moving target is known.

4.2.2 Brute Force Full-View Coverage Algorithm for a Moving Target

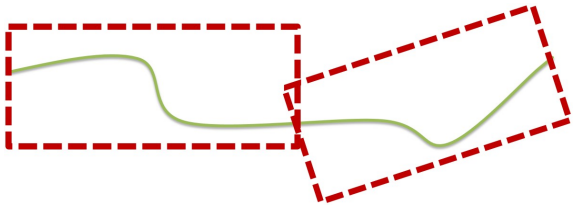

Figure 4.6: Divide the object's path into several segments

In order to minimize the number of camera sensors we develop a brute-force algorithm for a moving target consisting of three steps:

Step 1: Divide the path of the moving target into Multiple segments. For example, in Figure 4.6, two segments of the object's path are shown.

Step 2: For each segment, find the camera sensor set which full-view covers this segment.

Step 3: Perform computation to decide the active time duration for camera sensors deployed along each segment.

For the ith segment, its length ds can be obtained by

$$ds = \sqrt{x^{'}(t)^2 + y^{'}(t)^2}dt$$

The duration of active time is given by

$$dt = \frac{ds}{v_{imin}}$$

Here v_{imin} stands for the minimum speed of the object on ith segment. At the initial point of the current path segment, the camera sensor set are activated.At the ending point of the current path segment, the camera sensors except those also in the following camera sensor set should be deactivated.

Algorithm 8 Brute Force Full-View Coverage Algorithm for a Moving Target

Input: Cameras set C in the area of concern; the direction, position and speed of the moving target.
Output: a sequence of camera set $\{C_i, t_{ii}, t_{if}\}$, where C_i full-view cover the ith segment of path, t_{ii} is the initial moment to active C_i, and t_{if} is the moment to deactive C_i.

1: broadcast the information of the direction, position and speed of the moving target.
2: all camera sensors perform calculations to determine whether it should joint the monitoring scheme based on the shortest distance to the path exceeds the sensor range or not.
3: if yes, determine which segment it should monitor and the initial and final time to monitor.
4: Output the sequence of camera sensors $\{C_i, t_{ii}, t_{if}\}$.

Essentially, all camera sensors whose shortest distance to the path is within the sensing range will perform full-view coverage. The advantage of algorithm 8 is that it can guarantee the full-view covarage along the current segmemt, but possibly using too many camera sensors. As for active time there is room to reduce, obviously these camera sensors near the left side of the segment will be turned on for more time than needed. Hence we develop a heuristic algorithm to reduce the number of participating camera sensors and also minimize the time of participation of camera sensors.

4.2.3 Heuristic Full-View Coverage Algorithm for a Moving Target

In the brutal-force algorithm, excessive camera sensors are possibly exploited to achieve full-view coverage goal, and possibly the time duration for individual camera sensor is not minimum. Therefore, we develop an energy efficient distributing algorithm to minimize the number of camera sensors while guaranteeing the full–view coverage for the moving target and minimize the time duration for camera sensors involved in the monitoring process.

Theorem 4 *Given the effective angle θ_0 and sensing range R_s, suppose the minimum side length of the triangle is l_{min} in triangular –lattice based deployment which ensures the full-view coverage for the whole area, then if the maximum side length l of all triangles after perform Delaunay triangulation is not great than l_{min}, i.e., then the area is full-view covered by the existing camera sensor deployment.*

For example, in Figure 4.7, the left diagram is a triangle lattice-based area which is full-view covered, suppose in the right area the maximum side length of all triangles is not greater than the minimum side length of equilateral triangle, according to theory 4, the right area is full-view covered too.

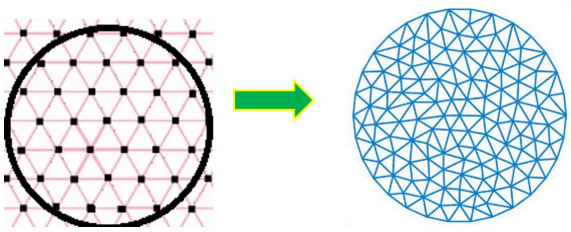

Figure 4.7: Illustration of Theory 4, if left area is full-view covered, then the right area is full-view covered too.

Based on Theorem 4, we describe the heuristic algorithm as follows:

Step 1: Conduct Delaunay triangulation for all camera sensors along the path of the object, where the sidelength of resultant triangles should be as close to the minimum side length of regular triangles l_{min} but not greater than it.

Step 2: along the path, each time the object passes a camera sensor closest to the path, update camera sensor cover.

Step 3: activate camera sensors newly entering the area of the circle centered at the camera sensor closest to the path with the radius of sensing range.

Step 4: deactivate camera sensors which moves out the circle.

This heuristic algorithm can achieve minimizing both the number of camera sensors involved in full-view coverage and the time duration of activation.

In conclusion, we develop effective algorithms to minimize the energy during object tracking process, hence prolong the lifetime for camera sensors.

Chapter 5

On Multi-dimension Full-View Coverage

Generally, it is assumed that all camera sensors reside on a plane in the 2D design of terrestrial camera sensor networks. In some applications the camera networks are distributed over a 3D space, e.g., camera sensors are required to be placed on the trees of different heights in a forest. 3D design is the obvious choice for monitoring railway tunnels or underground tunnels in the mine. For this reason, we address the 3D full-view coverage problem in camera sensor networks.

5.1 3-D Full-View Coverage Theory

In this section, we first develop a novel 3D full-view coverage model. Then we investigate the low bound problem of camera sensor numbers necessary to full-view cover a point. Next, we propose the 3D full-view coverage detection methodologies for an arbitrary point P. Finally, we tackle the 3D full-view coverage optimization problem in order to minimize the camera sensor number.

5.1.1 NOTATIONS AND MODEL

Suppose we deploy camera sensors set C to monitor a target P. Each camera sensor c in C can be modeled as shown in Figure 5.1 (a), where R stands for the sensing range, ϕ stands for the field-of-view (FoV) angle and $\vec{V}$ stands for the orientation vector. The spherical cone is a portion of sphere defined by a conical boundary with the apex at the center c (c is denoted as both the camera sensor and the center of the sphere) of the sphere, where the radius is R. For any two points P_1 and P_2, $\|P_1P_2\|$ represents the Euclidean distance between P_1 and P_2. Given any two vectors $\vec{v_1}$ and $\vec{v_2}$, $\alpha(\vec{v_1}, \vec{v_2})$ stands for the angle between $\vec{v_1}$ and $\vec{v_2}$, which is in range $[0, \pi]$. Any object P can be covered by c provided P is inside the spherical cone of c, i.e., $\|Pc\| \leq R$ and $\alpha(\vec{V}, c\vec{P}) \leq \phi/2$, where $c\vec{P}$ is the vector from c to P.

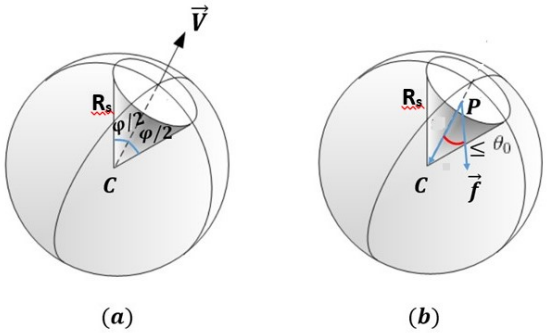

Figure 5.1: The 3D full-view coverage model.

Definition 7 *(3D Full-View Coverage) A target P is full-view covered if for any P's facing direction $\vec{f}$, there is a camera sensor c, such that P is covered by c and $\alpha(\vec{f}, \vec{Pc}) \leq \theta_0$, as shown in Figure 5.1 (b). Here $\theta_0 \in [0, \pi/2)]$ is a predefined value which is called the effective angle. A space is 3-D full-view covered if any point in this space is full-view covered.*

In comparison, sometimes we concern only some specific facing directions to be view covered, so we propose the partial-view coverage concept as follows:

Definition 8 *(3D Partial-View Coverage) A target P is partial-view covered if for any P's facing direction $\vec{f}$ within a cone apexed at P , there is a camera sensor c, such that P is covered by c and $\alpha(\vec{f}, \vec{Pc}) \leq \theta_0$.*

Lemma 4 *Suppose the total solid angle with respect to the point P $\Omega = \sum_{i=1}^{n} \Omega_i$, if Ω_i is partial-view covered, then the point P will be full-view covered.*

Proof: Based on Definition 7, if all solid angles are partially view covered, then for any facing directions of P there will be a camera sensor c such that P is covered by c and $\alpha(\vec{f}, \vec{Pc}) \leq \theta_0$.

In the following analysis, we assume that $\phi = 2\pi$, which can be implemented by bundling multiple camera sensors together to form a panoramic view or by rotating the camera sensor around if the rotation time is negligible. We make such a simplicity of analysis because the major challenge of $3D$ full-view coverage is due to the introduction of effective angle θ_0 instead of ϕ. We can extend the analysis for any $0 \leq \phi \leq 2\pi$ without loss of generality.

5.1.2 The Lower Bound of Number of Camera Sensors to Full-View Cover a Point

For any camera sensor $c_i \in C$, it can be guaranteed that the spherical cone centered at P and with a radius of R_s is partially view covered, where for some particular facing directions of point P, the angle between viewing direction of camera $c_i \in C$ and facing direction of point P is equal to or less than the effective angle θ_0. We use the term θ_0-view covered spherical cone to describe this situation.

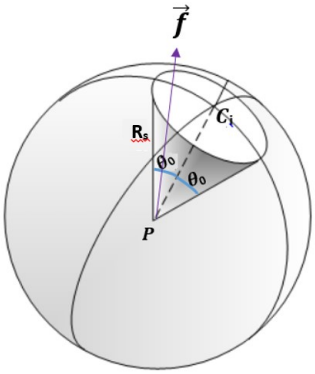

Figure 5.2: The θ_0-view covered spherical cone.

Definition 9 *(θ_0-view covered spherical cone) A target P is θ_0-view covered by a camera c_i if P's facing direction $\vec{f}$ is always within the solid angle apexed at P and with opening angle $2\theta_0$.*

We utilize the solid angle to represent the θ_0-view covered spherical cone. A solid angle (denoted by Ω) is a $3D$ angle subtended by a surface at a point P by joining its periphery points to that point. A solid angle is a measure of how large the object appears to an observer looking from that point. In Figure 5.2, the corresponding solid angle is the three-dimensional angle enclosed by the lateral surface of a cone at its vertex P. By using spherical coordinates, the definition of solid angle can be represented as:

$\Omega = \iint_S sin\phi d\theta d\phi$

where S denotes an integral over the surface of the projection, ϕ is the colatitude, and θ is the longitude. Consequently, the solid angle for Figure 5.2 $\Omega = 2\pi(1 - cos\theta_0)$.

Lemma 5 *Given the effective angle θ_0, the lower bound required to full-view cover a point P is $\lceil 2/(1 - cos\theta_0) \rceil$.*

Proof: the solid angle for the sphere subtended at the point P is 4π. For each camera sensor, it can partially view cover the point P by a solid angle $\Omega = 2\pi(1 - cos\theta_0)$. So the lower bound of the number of camera sensors is given by $\lceil 4\pi/(2\pi(1 - cos\theta_0)) \rceil = \lceil 2/(1 - cos\theta_0) \rceil$.

Figure 5.3 shows the relationship between the effective angle θ_0 and the lower bound of number of camera sensors to full-view cover the point P. In comparison to the 2D scenario, where the lower bound equals $\lceil \pi/\theta_0 \rceil$, the number of camera sensors in 3D space required for full-view covering a point is much greater if the effective angle is small. For example, when $\theta_0 = 20^o$, the lower bound for 2D is roughly 10 camera sensors in comparison with 35 camera sensors for 3D case. If the effective angle becomes even smaller, the difference increases dramatically. On the other hand, when the effective angle is relatively larger ($\theta_0 \geq 60^o$), the lower bound for 2D and 3D cases are approximately the same.

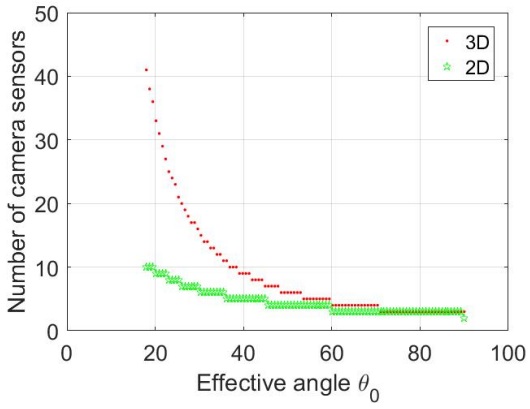

Figure 5.3: The lower bound of the number of camera sensors to full-view cover a point.

5.1.3 3D Full-View Coverage Detection Methodologies

Given a deployed target field, how to determine that an arbitrary target P is 3D full-view covered or not? To address this problem, we first generate a camera sensor set C where the point P is within the sensing range R_s of camera sensor $c_i \in C$, whereas other camera sensors do not involve in the full-view coverage process for P.

Next, in order to simplify the 3D full-view coverage detection process, we move all camera sensors $\in C$ to a unit sphere without loss of 3D full-view coverage validity. Finally, we demonstrate our 3D full-view coverage detection methodology.

Construction of a Unit Sphere

Since all camera sensors participating the full-view coverage process should be within the sensing range R_s, all camera sensors should be located inside a sphere with a radius of R_s and centered at P. Based on the Definition 7, the distance from each camera sensor to the point P will not affect the full-view coverage validity, that is, if we maintain the orientation of each camera sensor and move the camera sensor along the radius where it is currently located, the full-view coverage validity will remain unchanged. Consequently, we can move each camera sensor along its current radius to the surface of the sphere with radius of R_s and centered at the point P. Furthermore, we can normalize the sensor range R_s to a unit such that we obtain a unit sphere, where each camera sensor is distributed on the unit sphere.

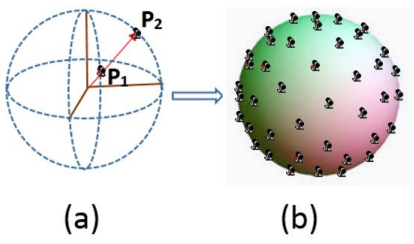

Figure 5.4: Construction of a unit sphere.

As shown in figure 5.4 (a), suppose an arbitrary camera sensor is currently located at P_1. While maintaining its orientation we can move it from P_1, which is inside the sphere with the radius of R_s and centered of P, to P_2, which is located on the surface of the sphere. We repeat the same process for the next camera sensor until all other camera sensors are on the surface of the sphere. Next, we normalize the radius of R_s to 1 to create a unit sphere as demonstrated in figure 5.4 (b), where all camera sensors are moved to the surface of a unit sphere while conserving full-view coverage validity for P. For next step we consider just the surface of the sphere, not its interior, in such a way we can simplify the 3D full-view coverage detection process.

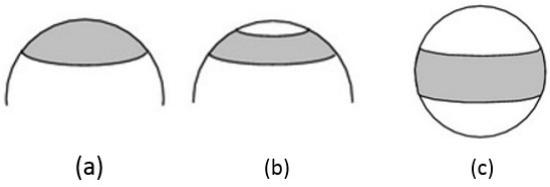

Figure 5.5: Divide the spherical surface into spherical belts, if each of which can be view-covered, then the spherical surface will be view-covered.

To check a point is full-view covered or not, we utilize the divide-and-conquer strategy as shown in Figure 5.5, where the above figure is a spherical cap. Obviously, the corresponding solid angle is θ_0 view covered. In the middle in Figure 5.5 there is a spherical belt if the low circle is uniformly deployed by camera sensors, then this belt will be θ_0 view covered. In the bottom in Figure 5.5, a similar conclusion can be drawn. So if we divide the spherical surface into spherical belts, and prove each belts is θ_0 view covered, there the whole sphere will be covered.

5.1.4 3D Full-View Coverage Deployment Optimization Problem

In this section, we consider a set of camera sensors indicated by C, where the point P is within the sensing range R_s of camera sensor $c_i \in C$ to cover the surface of a sphere and demonstrate the full-view coverage for a point is equivalent to cover a sphere with spherical caps of camera sensors. Then we prove the optimization problem for 3D full-view coverage can be viewed from either of two perspectives, the full-view coverage covering problem and the full-view coverage packing problem. The solution for full-view coverage packing problem can provide a lower bound to the solution of the full-view coverage covering problem. In other word, the full-view coverage packing and covering problem are dual problems.

The full-view coverage covering problem

Suppose given the effective angle θ_0, then what is the minimum number of camera sensors in order to full-view cover a point? In Figure 5.2, there exists a spherical cap corresponding to the spherical cone. Intuitively, the 3D full-view coverage problem for a point can be converted into the sphere covering problem with overlapping spherical caps, as shown in Figure 5.6.

The effective angle θ_0 equals the angular radius of the spherical cap, so the full-view coverage optimization problem, given the effective angle, how must a sphere be covered by n equal spherical caps so that the angular radius of the spherical caps will be as small as possible? This question is equivalent to the following mathematical problem: How must a sphere be covered by spherical caps with common angular radius θ_0 so that the number of spherical caps will be as small as possible? This problem has a dual counterpart in packing. This is the well-known Tammes problem: How must n equal non-overlapping spherical caps be packed on a sphere so that the angular radius of the spherical caps will be as large as possible?

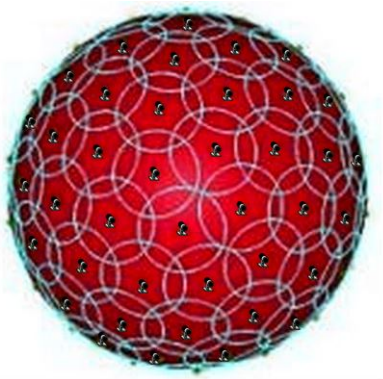

Figure 5.6: On the sphere, the center of each small circle stands for a camera sensor. The 3D full-view coverage problem for a point can be converted into the spherical cap covering problem

In figure 5.6, the object is located in the center of the sphere, each small circle stands for a spherical cap, and the camera sensor is located in the center of the spherical cap.

Given the number of camera sensors n and the effective angle θ_0 of each camera sensor in figure 5.6, which is equivalent to the angular radius of the spherical cap, we can calculate the full-view coverage density D_n and full-view coverage efficiency η_n which are defined as follows.

Definition 10 *(full-view coverage density) The full-view coverage density D_n is defined as the ratio of the total area of the surface of the spherical caps of camera sensors to the surface area of the unit sphere.*

To minimize the number of camera sensors to full-view cover a point is equivalent to find the minimal full-view coverage density.

Lemma 6 *The full-view coverage density $D_n = (1 - cos\theta_0)n/2$.*

Proof: In figure 5.2, the solid angle of spherical cone equals to the area of the corresponding spherical cap since the sphere is a unit one. For each camera sensor, the corresponding solid angle equals to $2\pi(1 - cos\theta_0)$, consequently the full coverage density in Figure 5.6 equal to $D_n = 2\pi(1 - cos\theta_0)n/(4\pi) = (1 - cos\theta_0)n/2$.

Definition 11 *(full-view coverage efficiency) The full-view coverage efficiency $\eta_n = 1/D_n$.*

Intuitively, in figure 5.6 $D_n > 1$, so $\eta_n = 2/((1 - cos\theta_0)n) < 1$. Therefore, the smaller the n, the higher full-view coverage efficiency we can achieve.

The full-view coverage packing problem

Given the effective angle θ_0, what is minimum value of camera sensor number n in order to achieve 3D full-view coverage for a point P?

In figure 5.6, imagine we decrease progressively the effective angle θ_0, the spherical caps will decrease too and ultimately we can expect that the overlap areas on the spherical surface will shrink to some points as shown in figure 5.7, where we denote the corresponding effective angle as $\theta_0^{'}$. In figure 5.7, each dot on the unit sphere stands for a camera sensor, each circle represents for a spherical cone of camera sensor. Though in figure 5.7 the point is not full-view covered, we can progressively increase the effective angle $\theta_0^{'}$ until each intersection is enlarged to cover its adjacent holes and the effective angle equals to θ_0, where the point P will be full-view covered.

Definition 12 *(full-view coverage packing) The full-view coverage density D_n is defined as the ratio of the total area of the surface of the spherical caps of camera sensors to the surface area of the unit sphere.*

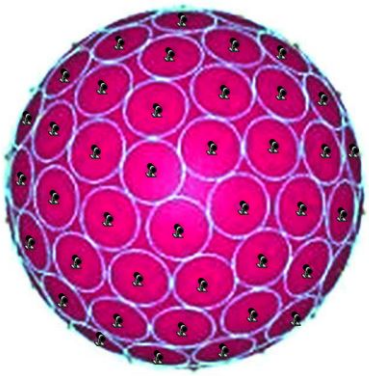

Figure 5.7: Spherical caps of camera sensor packing on a unit sphere

A system of equal spherical caps in 3-dimensional space is said to form a packing, if no two spherical caps of the system have any inner point in common, i.e., any point on the sphere belongs to at most one spherical cap. One the other hand, in the spherical cap covering problem [80, 81, 87, 82, 84, 83, 85, 86], any point on the sphere belongs to at least one spherical cap. Since the spherical cap covering and packing problem is a dual, we can further convert the spherical cap covering problem into the spherical cap packing problem[88, 98, 89, 96, 90, 92, 93, 94, 95, 97, 91]. Consider the spherical packing problem in Figure 5.8, for each spherical cap, there exists a unique sphere of radius r, where the spherical cap is the projection of the sphere. Therefore the 3-D full-view coverage problem is converted into the spherical packing problem corresponds to the deployment of n spheres with radius r around a central sphere with radius R_s. From simple trigonometry,

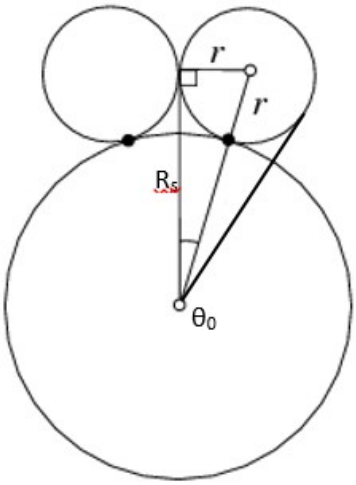

Figure 5.8: On the sphere, the center of each small circle stands for a camera sensor. The 3D full-view coverage problem for a point can be converted into the spherical cap covering problem

$$sin(\theta_0) = \frac{r}{r+R_s}$$

So the radii r of the n sphere are given by

$$r = \frac{R_s}{csc(\frac{1}{2}\theta_0)-1}$$

After normalizing the radius, we get the normalized radius of the n spheres

$$r' = \frac{1}{csc(\frac{1}{2}\theta_0)-1}$$

The corresponding central sphere is a unit sphere. The 3-D full-view coverage problem is converted into spherical packing one where the solution is known for $n \leq 130$ as shown in figure 5.9.

5.1.5 Minimum Camera Sensors for 3D Full-View Coverage Given Effective Angle

In table 5.1 [87], the first column stands for the interval of effective angle θ_0 and the second column represents the minimum number of camera sensors for 3D covering a target. For example, in the first row, when the effective angle $70.528779^o \leq \theta_0 < 90^o$, the minimum number of camera sensors is 4, which constitutes a tetrahedron on the surface of the unit sphere. When n=6, the effective angle $54.735610^o \leq \theta_0 < 63.434949^o$, the deployment of camera sensors on the unit sphere constitutes an octahedron. If the effective angle $48.139529^o \leq \theta_0 < 51.026553^o$, the minimum number of camera sensors is 8, which constitutes a cube on the surface of the unit sphere. When the effective angle $37.377368^o \leq \theta_0 < 41.427196^o$, the minimum number of camera sensors is 12, and the 12 camera sensors constitute an icosahedron on the unit sphere. If the effective angle $29.623096^o \leq \theta_0 < 30.382284^o$, the minimum number of camera sensors is 20, which constitutes a dodecahedron on the surface of the unit sphere.

Table 5.1: Minimum Camera Number for Given Effective Angle

θ_0 interval	Num_{min}
$[70.528779^o, 90^o)$	4
$[63.434949^o, 70.528779^o)$	5
$[54.735610^o, 63.434949^o)$	6
$[51.026553^o, 54.735610^o)$	7
$[48.139529^o, 51.026553^o)$	8
$[45.878888^o, 48.139529^o)$	9
$[42.307827^o, 45.878888^o)$	10
$[41.427196^o, 42.307827^o)$	11
$[37.377368^o, 41.427196^o)$	12
$[37.068543^o, 37.377368^o)$	13
$[34.937927^o, 37.068543^o)$	14
$[34.039900^o, 34.937927^o)$	15
$[32.898812^o, 34.039900^o)$	16
$[32.092933^o, 32.898812^o)$	17
$[31.013172^o, 32.092933^o)$	18
$[30.382284^o, 31.013172^o)$	19
$[29.623096^o, 30.382284^o)$	20

In Figure 5.9, we demonstrate the required minimum number of camera sensors corresponding to the effective angle interval $11.3165625^o \leq \theta_0 < 90^o$. Since normally the effective angle $30^o \leq \theta_0 < 90^o$, this result is good enough for practical purposes to deploy 3D camera sensor networks. The whole effective range from 11.3165625^o to 90^o is divided into a serious of subrange, each of which corresponds to a minimum number of camera sensor for 3D full-view covering a target [99]. When the effective angle decreases, the minimum number of camera sensors increases dramatically.

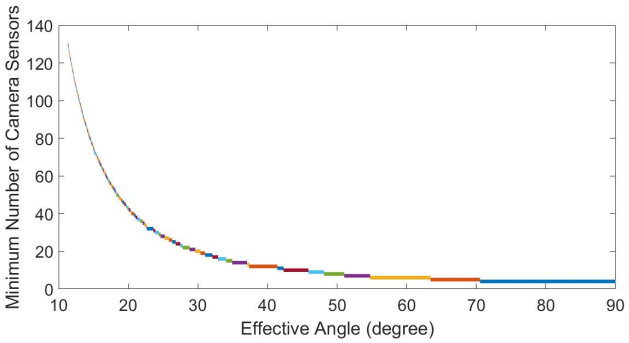

Figure 5.9: The effective angle θ_0 vs the minimal number of camera sensors to 3D full-view cover a point P.

In Figure 5.10, the covering density is shown. When the effective angle approaches to 90^o, the spherical cap corresponding to one camera sensor is roughly a semi-sphere. Since there are four camera sensors, we get four semi-spheres and that is why the covering density equals to 2. This figure is based on the covering density formula is $D_n = (1 - cos\theta_0)n/2$. When the effective angle is larger, the covering density is relatively greater. When the effective angle becomes very small, the covering density approaches to a stable value, which is approximately 1.22.

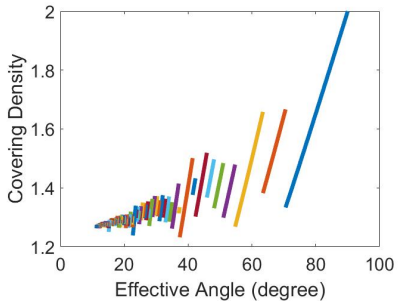

Figure 5.10: The effective angle θ_0 vs covering density.

5.1.6 Conclusions

In this section, we propose a novel 3D full-view coverage model and investigate the low bound to full-view cover a point. We also demonstrate the methodologies for 3D full-view coverage detection problem. Finally, we tackle the full-view coverage optimization problem in order to minimize the number of camera sensors and demonstrate a valid solution.

Chapter 6

Conclusions and Future Work

6.1 Summary

In this dissertation, we have studied the coverage problems in camera sensor networks. First, a novel analytical coverage model called full-view coverage has been proposed, which characterizes unique properties of camera sensors in terms of coverage. With this model, we proposed a series of methods for full-view coverage detection, and derived sufficient conditions on the number of camera sensors needed for full-view coverage in a controlled triangular lattice based deterministic deployment. Theoretical analysis has proved the correctness of the detection algorithm. we also propose the k-full-view coverage problem where k disjoint subsets of cameras are able to provide full-view coverage for a target or a given area. We prove that flaws exists in the current full-view coverage theory, and propose a new full-view coverage analytic theory, which is simpler, efficient and more feasible for practical applications than the previous complex geometric counterpart. We address the minimization problem of full-view coverage camera sensor set, which aims to maximize the number of disjoint camera subsets for k-full-view coverage. We develop the k-coverage algorithms both for a target and for given area based on equilateral triangle displacement. We have studied other problems such as 3-D Full-view coverage, object tracking problem and energy-awareness problem in full-view coverage camera sensor networks.

6.2 Future Work

There are many interesting works worth further study. One potential problem is the partial-view coverage problem where the cameras only partially cover the object's facets. Dependent on the percentage of the surface of the object needed to cover, we can estimate the number of sensors needed to achieve partial-view coverage. A related problem is regarding the relationship between the portion of partial-view covered area and the number of deployed sensors. For example, the system designer may ask for an

estimate of the sensors needed to achieve full-view coverage over 50 percent area (rather than over the whole area).

Some other interesting works can be done following up the k-full-view coverage problem is to develop energy awareness algorithm. Our research on this issue is still at initial stage. So there are still many interesting and practical issues to be further investigated.

We can extend the full-view coverage concept to full-*-coverage. Here * stands for infrared light, ultra sound, etc., where information can be extracted by the infrared sensors [100], ultra sound sensors [101]. Like camera sensors, infrared sensors and ultra sound sensors are not isotropic sensors which can equally detect the environment in each orientation. They have a limited angle of sensing range due to special applications. Therefore, the fundamental insights of camera sensors can be applied to these similar situations involved with infrared or ultra sound sensors.

Bibliography

[1] Y. Wang and G. Cao, "On full-view coverage in camera sensor networks," in *INFOCOM, 2011 Proceedings IEEE.* IEEE, 2011, pp. 1781–1789.

[2] G. S. Kasbekar, Y. Bejerano, and S. Sarkar, "Lifetime and coverage guarantees through distributed coordinate-free sensor activation," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 2, pp. 470–483, 2011.

[3] Y. Cai, W. Lou, M. Li, and X.-Y. Li, "Energy efficient target-oriented scheduling in directional sensor networks," *Computers, IEEE Transactions on*, vol. 58, no. 9, pp. 1259–1274, 2009.

[4] C. Sanderson, T. Shang, and B. C. Lovell, "Towards pose-invariant 2d face classification for surveillance," in *Analysis and Modeling of Faces and Gestures.* Springer, 2007, pp. 276–289.

[5] X. Gao, R. Yang, F. Wu, G. Chen, and J. Zhou, "Optimization of full-view barrier coverage with rotatable camera sensors," in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on.* IEEE, 2017, pp. 870–879.

[6] Y. Hong, R. Yan, Y. Zhu, D. Li, and W. Chen, "Finding best and worst-case coverage paths in camera sensor networks for complex regions," *Ad Hoc Networks*, vol. 56, pp. 202–213, 2017.

[7] S. B. B. Priyadarshini and S. Panigrahi, "A distributed approach based on hierarchical scalar leader selection for enhanced event coverage in wireless multimedia sensor networks," in *Distributed Computing and Internet Technology.* Springer, 2017, pp. 3–14.

[8] Y. Hong, D. Li, D. Kim, W. Chen, J. Yu, and A. O. Tokuta, "Maximizing target-temporal coverage of mission-driven camera sensor networks," *Journal of Combinatorial Optimization*, pp. 1–23, 2017.

[9] J. Zeng, T. Wang, and M. Bhuiyan, "Cooperative relay tracking with multiple mobile camera sensors," in *Proceedings of the ACM Turing 50th Celebration Conference-China.* ACM, 2017, p. 42.

[10] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad-hoc sensor networks," *Computer communications*, vol. 29, no. 4, pp. 413–420, 2006.

[11] S. Kumar, T. H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *Proceedings of the 10th annual international conference on Mobile computing and networking.* ACM, 2004, pp. 144–158.

[12] P.-J. Wan and C.-W. Yi, "Coverage by randomly deployed wireless sensor networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2658–2669, 2006.

[13] V. Blanz, P. Grother, J. P. Phillips, and T. Vetter, "Face recognition based on frontal views generated from non-frontal images," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 454–461.

[14] P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott, and M. Sharpe, "Frvt 2006 and ice 2006 large-scale results," *National Institute of Standards and Technology, NISTIR*, vol. 7408, p. 1, 2007.

[15] G. Wang, G. Cao, and T. La Porta, "Proxy-based sensor deployment for mobile sensor networks," in *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on.* IEEE, 2004, pp. 493–502.

[16] W. Li and Y. Wu, "Tree-based coverage hole detection and healing method in wireless sensor networks," *Computer Networks*, vol. 103, pp. 33–43, 2016.

[17] X. Du, M. Zhang, K. E. Nygard, S. Guizani, and H.-H. Chen, "Self-healing sensor networks with distributed decision making," *International Journal of Sensor Networks*, vol. 2, no. 5-6, pp. 289–298, 2007.

[18] S. Babaie and S. S. Pirahesh, "Hole detection for increasing coverage in wireless sensor network using triangular structure," *arXiv preprint arXiv:1203.3772*, 2012.

[19] X. Li, D. K. Hunter, and K. Yang, "Wlc12-1: Distributed coordinate-free hole detection and recovery," in *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE.* IEEE, 2006, pp. 1–5.

[20] W. Li and W. Zhang, "Coverage hole and boundary nodes detection in wireless sensor networks," *Journal of network and computer applications*, vol. 48, pp. 35–43, 2015.

[21] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[22] C. Qiu, H. Shen, and K. Chen, "An energy-efficient and distributed cooperation mechanism for k-coverage hole detection and healing in wsns," in *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on.* IEEE, 2015, pp. 73–81.

[23] M. R. Senouci, A. Mellouk, and K. Assnoune, "Localized movement-assisted sensordeployment algorithm for holedetection and healing," *IEEE Transactions on parallel and distributed systems*, vol. 25, no. 5, pp. 1267–1277, 2014.

[24] H. Trigui, "Coverage hole detector," Dec. 1 2009, uS Patent App. 12/628,442.

[25] M. Argany, M. A. Mostafavi, and F. Karimipour, "Voronoi-based approaches for geosensor networks coverage determination and optimisation: A survey," in *Voronoi Diagrams in Science and Engineering (ISVD), 2010 International Symposium on.* IEEE, 2010, pp. 115–123.

[26] A. Galindo-Serrano, B. Sayrac, S. B. Jemaa, J. Riihijärvi, and P. Mähönen, "Automated coverage hole detection for cellular networks using radio environment maps," in *Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks (WiOpt), 2013 11th International Symposium on.* IEEE, 2013, pp. 35–40.

[27] J. Yao, G. Zhang, J. Kanno, and R. Selmic, "Decentralized detection and patching of coverage holes in wireless sensor networks," in *Proc. of SPIE*, vol. 7352, 2009, p. 73520V.

[28] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4. IEEE, 2005, pp. 2302–2312.

[29] D. Anand, H. Chandrakanth, M. Giriprasad, and A. Pradesh, "Energy efficient coverage problems in wireless ad hoc sensor networks," *ACJ Journal*, vol. 2, no. 2, pp. 42–50, 2011.

[30] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda *et al.*, "A line in the sand: a wireless sensor network for target detection, classification, and tracking," *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.

[31] N. Ahmed, S. S. Kanhere, and S. Jha, "The holes problem in wireless sensor networks: a survey," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 2, pp. 4–18, 2005.

[32] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, 2005.

[33] Y. Bejerano, "Simple and efficient k-coverage verification without location information," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE.* IEEE, 2008.

[34] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad hoc networks*, vol. 7, no. 3, pp. 537–568, 2009.

[35] B. Cărbunar, A. Grama, J. Vitek, and O. Cărbunar, "Redundancy and coverage detection in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 1, pp. 94–128, 2006.

[36] H. Zhang and J. Hou, "On deriving the upper bound of α-lifetime for large sensor networks," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing.* ACM, 2004, pp. 121–132.

[37] P. Balister, Z. Zheng, S. Kumar, and P. Sinha, "Trap coverage: Allowing coverage holes of bounded diameter in wireless sensor networks," in *INFOCOM 2009, IEEE.* IEEE, 2009, pp. 136–144.

[38] P. Balister, B. Bollobas, A. Sarkar, and S. Kumar, "Reliable density estimates for coverage and connectivity in thin strips of finite length," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking.* ACM, 2007, pp. 75–86.

[39] B. Liu, O. Dousse, J. Wang, and A. Saipulla, "Strong barrier coverage of wireless sensor networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing.* ACM, 2008, pp. 411–420.

[40] R. Kershner, "The number of circles covering a set," *American Journal of mathematics*, vol. 61, no. 3, pp. 665–671, 1939.

[41] T.-J. Chang, K. Wang, and Y.-L. Hsieh, "A color-theory-based energy efficient routing algorithm for mobile wireless sensor networks," *Computer Networks*, vol. 52, no. 3, pp. 531–541, 2008.

[42] F. Li, J. Barabas, and A. L. Santos, "Information processing for live photo mosaic with a group of wireless image sensors," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks.* ACM, 2010, pp. 58–69.

[43] Y. Wang and G. Cao, "Achieving full-view coverage in camera sensor networks," *ACM Transactions on Sensor Networks (ToSN)*, vol. 10, no. 1, p. 3, 2013.

[44] X. Bai, Z. Yun, D. Xuan, T. H. Lai, and W. Jia, "Optimal patterns for four-connectivity and full coverage in wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 3, pp. 435–448, 2010.

[45] S. Fleck, F. Busch, P. Biber, and W. Straber, "3d surveillance a distributed network of smart cameras for real-time tracking and its visualization in 3d," in *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06).* IEEE, 2006, pp. 118–118.

[46] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, "Mesheye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," in *Proceedings of the 6th international conference on Information processing in sensor networks.* ACM, 2007, pp. 360–369.

[47] H. Pasula, S. Russell, M. Ostland, and Y. Ritov, "Tracking many objects with many sensors," in *IJCAI*, vol. 99, 1999, pp. 1160–1171.

[48] E. Monari and K. Kroschel, "A knowledge-based camera selection approach for object tracking in large sensor networks," in *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*. IEEE, 2009, pp. 1–8.

[49] A. Brooks and S. Williams, "Tracking people with networks of heterogeneous sensors," in *Proceedings of the Australasian Conference on Robotics and Automation*. Citeseer, 2003, pp. 1–7.

[50] Y. Wang and G. Cao, "Barrier coverage in camera sensor networks," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2011, p. 12.

[51] A. Saipulla, B. Liu, G. Xing, X. Fu, and J. Wang, "Barrier coverage with sensors of limited mobility," in *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2010, pp. 201–210.

[52] S. He, J. Chen, X. Li, X. Shen, and Y. Sun, "Cost-effective barrier coverage by mobile sensor networks," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 819–827.

[53] D. Tao, S. Tang, H. Zhang, X. Mao, and H. Ma, "Strong barrier coverage in directional sensor networks," *Computer Communications*, vol. 35, no. 8, pp. 895–905, 2012.

[54] A. Chen, T. H. Lai, and D. Xuan, "Measuring and guaranteeing quality of barrier-coverage in wireless sensor networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2008, pp. 421–430.

[55] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 28–39.

[56] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, 2008.

[57] K. Sohraby, D. Minoli, and T. Znati, *Wireless sensor networks: technology, protocols, and applications*. John Wiley & Sons, 2007.

[58] B. Wang, W. Wang, V. Srinivasan, and K. C. Chua, "Information coverage for wireless sensor networks," *IEEE Communications Letters*, vol. 9, no. 11, pp. 967–969, 2005.

[59] W. W. V. Srinivasan and K.-C. Chua, "Trade-offs between mobility and density for coverage in wireless sensor networks," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM, 2007, pp. 39–50.

[60] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 2, pp. 262–267, 2011.

[61] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *INFOCOM 2005. 24th annual joint conference of the ieee computer and communications societies. proceedings ieee*, vol. 3. IEEE, 2005, pp. 1976–1984.

[62] K. Romer and F. Mattern, "The design space of wireless sensor networks," *IEEE wireless communications*, vol. 11, no. 6, pp. 54–61, 2004.

[63] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[64] S. Tozlu, M. Senel, W. Mao, and A. Keshavarzian, "Wi-fi enabled sensors for internet of things: A practical approach," *IEEE Communications Magazine*, vol. 50, no. 6, 2012.

[65] R. D. Komguem, R. Stanica, M. Tchuente, and F. Valois, "Node ranking in wireless sensor networks with linear topology," in *Wireless Days, 2017*. IEEE, 2017, pp. 127–132.

[66] Z. Tonggang, S. Ke, and P. Dafa, "Design and realization for the protocol of linear long-distance wireless sensor network," in *Instrumentation & Measurement, Computer, Communication and Control (IMCCC), 2016 Sixth International Conference on*. IEEE, 2016, pp. 61–64.

[67] R. Choudharyl, S. Kumar, A. Deepak, and D. Dash, "Data aggregation in wireless sensor network: An integer linear programming formulation for energy optimization," in *Wireless and Optical Communications Networks (WOCN), 2016 Thirteenth International Conference on*. IEEE, 2016, pp. 1–6.

[68] X. Lv, J. Hao, X. Jia, Z. Han, and B. Yang, "Optimal power control based opportunistic routing in linear wireless sensor networks," in *Control Conference (CCC), 2016 35th Chinese*. IEEE, 2016, pp. 8402–8407.

[69] F. Alduraibi, N. Lasla, and M. Younis, "Coverage-based node placement optimization in wireless sensor network with linear topology," in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[70] T.-V. Poh, J. Jiang, and M. J. Reed, "Multicommodity flow optimization in support for packet-switched network traffic engineering," in *IP Operations and Management, 2004. Proceedings IEEE Workshop on*. IEEE, 2004, pp. 23–28.

[71] N. E. Young, "Sequential and parallel algorithms for mixed packing and covering," in *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on.* IEEE, 2001, pp. 538–546.

[72] L. K. Fleischer, "Approximating fractional multicommodity flow independent of the number of commodities," *SIAM Journal on Discrete Mathematics*, vol. 13, no. 4, pp. 505–520, 2000.

[73] Y. Xue, Y. Cui, and K. Nahrstedt, "Maximizing lifetime for data aggregation in wireless sensor networks," *Mobile Networks and Applications*, vol. 10, no. 6, pp. 853–864, 2005.

[74] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Efficient energy management in sensor networks," *Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing*, vol. 2, pp. 71–90, 2005.

[75] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[76] J. Elson and K. Römer, "Wireless sensor networks: A new regime for time synchronization," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 149–154, 2003.

[77] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng, "Efficient in-network moving object tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, pp. 1044–1056, 2006.

[78] Y. Xu, J. Winter, and W.-C. Lee, "Prediction-based strategies for energy saving in object tracking sensor networks," in *Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on.* IEEE, 2004, pp. 346–357.

[79] L. Doherty, L. El Ghaoui *et al.*, "Convex position estimation in wireless sensor networks," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2001, pp. 1655–1663.

[80] I. Dumer, "Covering spheres with spheres," *Discrete & Computational Geometry*, vol. 38, no. 4, pp. 665–679, 2007.

[81] C.-F. Huang, Y.-C. Tseng, and L.-C. Lo, "The coverage problem in three-dimensional wireless sensor networks," in *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, vol. 5. IEEE, 2004, pp. 3182–3186.

[82] J. Bourgain and J. Lindenstrauss, "On covering a set in rd by balls of the same diameter," *Lecture Notes in Math*, vol. 1469, pp. 138–144, 1991.

[83] A. D. Wyner, "Random packings and coverings of the unit n-sphere," *The Bell System Technical Journal*, vol. 46, no. 9, pp. 2111–2118, 1967.

[84] J.-L. Verger-Gaugry, "Covering a ball with smaller equal balls in r n," *Discrete & Computational Geometry*, vol. 33, no. 1, pp. 143–155, 2005.

[85] B. Metzler and R. Pail, "Goce data processing: the spherical cap regularization approach," *Studia Geophysica et Geodaetica*, vol. 49, no. 4, pp. 441–462, 2005.

[86] H. T. Croft, K. Falconer, and R. K. Guy, *Unsolved Problems in Geometry: Unsolved Problems in Intuitive Mathematics*. Springer Science & Business Media, 2012, vol. 2.

[87] T. Tarnai and Z. Gáspár, "Covering a sphere by equal circles, and the rigidity of its graph," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 110, no. 1. Cambridge University Press, 1991, pp. 71–89.

[88] F. M. Richards, "Areas, volumes, packing, and protein structure," *Annual review of biophysics and bioengineering*, vol. 6, no. 1, pp. 151–176, 1977.

[89] E. B. Saff and A. B. Kuijlaars, "Distributing many points on a sphere," *The mathematical intelligencer*, vol. 19, no. 1, pp. 5–11, 1997.

[90] A. M. Odlyzko and N. J. Sloane, "New bounds on the number of unit spheres that can touch a unit sphere in n dimensions," *Journal of Combinatorial Theory, Series A*, vol. 26, no. 2, pp. 210–214, 1979.

[91] E. A. Rakhmanov, E. Saff, and Y. Zhou, "Minimal discrete energy on the sphere," *Math. Res. Lett*, vol. 1, no. 6, pp. 647–662, 1994.

[92] T. C. Hales, "The sphere packing problem," *Journal of Computational and Applied Mathematics*, vol. 44, no. 1, pp. 41–76, 1992.

[93] F. Pfender and G. M. Ziegler, "Kissing numbers, sphere packings, and some unexpected proofs," *Notices-American Mathematical Society*, vol. 51, pp. 873–883, 2004.

[94] T. W. Melnyk, O. Knop, and W. R. Smith, "Extremal arrangements of points and unit charges on a sphere: equilibrium configurations revisited," *Canadian Journal of Chemistry*, vol. 55, no. 10, pp. 1745–1761, 1977.

[95] J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices and groups*. Springer Science & Business Media, 2013, vol. 290.

[96] J. H. Conway, R. H. Hardin, and N. J. Sloane, "Packing lines, planes, etc.: Packings in grassmannian spaces," *Experimental mathematics*, vol. 5, no. 2, pp. 139–159, 1996.

[97] G. F. Tóth, "New results in the theory of packing and covering," in *Convexity and its Applications*. Springer, 1983, pp. 318–359.

[98] H. Coxeter, "Division of mathematics: The problem of packing a number of equal nonoverlapping circles on a sphere," *Transactions of the New York Academy of Sciences*, vol. 24, no. 3 Series II, pp. 320–331, 1962.

[99] R. Graham and N. Sloane, "On the covering radius of codes," *IEEE Transactions on Information Theory*, vol. 31, no. 3, pp. 385–401, 1985.

[100] P. Chen, S. Oh, M. Manzo, B. Sinopoli, C. Sharp, K. Whitehouse, O. Tolle, J. Jeong, P. Dutta, J. Hui *et al.*, "Instrumenting wireless sensor networks for real-time surveillance," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on.* IEEE, 2006, pp. 3128–3133.

[101] J. Li, R.-c. Wang, H.-p. Huang, and L.-j. Sun, "Voronoi based area coverage optimization for directional sensor networks," in *Electronic Commerce and Security, 2009. ISECS'09. Second International Symposium on*, vol. 1. IEEE, 2009, pp. 488–493.